# Contents

1 Introduction
   1.1 About this publication .................................................. 1
   1.2 Preface ............................................................................ 2
   1.3 Application areas ............................................................. 2
      1.3.1 Computational fluid dynamics ........................................... 3
      1.3.2 The model system ....................................................... 3
      1.3.3 Release and dispersion ............................................... 4
      1.3.4 Gas explosions ......................................................... 4
      1.3.5 Hydrogen safety ....................................................... 5
      1.3.6 Dust explosions ....................................................... 5
      1.3.7 Blast wave propagation .............................................. 5
      1.3.8 Future developments ................................................ 6
      1.3.9 Validation and documentation ...................................... 6
   1.4 Acknowledgements .......................................................... 6
   1.5 About this manual ............................................................ 8
      1.5.1 Printing conventions in this manual ................................. 8
      1.5.2 Special messages ..................................................... 9
      1.5.3 Job numbers .......................................................... 9
   1.6 What is new in this release? ................................................ 10
      1.6.1 FLACS-CFD 22.2 ....................................................... 10
         1.6.1.1 Simulators ......................................................... 10
         1.6.1.2 FGC ............................................................. 10
         1.6.1.3 CASD .......................................................... 10
         1.6.1.4 Flowvis ......................................................... 11
         1.6.1.5 RunManager .................................................. 11
      1.6.2 FLACS-CFD 22.1r2 .................................................... 12
         1.6.2.1 CASD .......................................................... 12
         1.6.2.2 Flowvis ......................................................... 12
      1.6.3 FLACS-CFD 22.1 ....................................................... 12
         1.6.3.1 Simulators ......................................................... 12
         1.6.3.2 CASD .......................................................... 12
         1.6.3.3 Flowvis ......................................................... 13
         1.6.3.4 RunManager .................................................. 13
      1.6.4 FLACS-CFD 21.3 ....................................................... 13
         1.6.4.1 Simulators ......................................................... 13
         1.6.4.2 CASD .......................................................... 14
         1.6.4.3 Flowvis ......................................................... 14
      1.6.5 FLACS-CFD 21.2 ....................................................... 14
         1.6.5.1 Simulators ......................................................... 14
         1.6.5.2 CASD .......................................................... 15
         1.6.5.3 Flowvis ......................................................... 15
      1.6.6 FLACS-CFD 21.1 ....................................................... 15
1.6.7 FLACS-CFD 20.2 ................................................................. 16
1.6.8 FLACS-CFD 20.1 ................................................................. 16
  1.6.8.1 Simulators ............................................................... 17
  1.6.8.2 CASD ................................................................. 17
  1.6.8.3 Flowvis ................................................................. 17
  1.6.8.4 FLACS-Risk ......................................................... 18
  1.6.8.5 RunManager .......................................................... 18
  1.6.8.6 Other ................................................................. 18

1.7 Compatibility ................................................................. 18
  1.7.1 Compatibility between FLACS-CFD 22.2 and 22.1 and 21.x .. 18
  1.7.2 Compatibility between FLACS-CFD 21.x and 20.2 .............. 19
  1.7.3 Compatibility between FLACS-CFD 20.1 and 20.2 .............. 19
  1.7.4 Compatibility between FLACS v10.9 and FLACS-CFD 20.1 .... 19
  1.7.5 Compatibility between FLACS v10.8 and v10.9 ................. 19
  1.7.6 Compatibility between FLACS v10.7 and v10.8 ................. 20
  1.7.7 Compatibility between FLACS v10.6 and v10.7 ................. 20
  1.7.8 Compatibility between FLACS v10.5 and v10.6 ................. 20
  1.7.9 Compatibility between FLACS v10.4 and v10.5 ................. 20
  1.7.10 Compatibility between FLACS v10.4r1 and v10.4r2 ......... 20
  1.7.11 Compatibility between FLACS v10.3 and v10.4 ............... 21
    1.7.11.1 Changes in results due to bug-fixes in Porcalc .......... 21
  1.7.12 Compatibility between FLACS v10.2 and v10.3 ............... 21
  1.7.13 Compatibility between FLACS v10.1 and v10.2 ............... 22
  1.7.14 Compatibility between FLACS v10.0 and v10.1 ............... 22
  1.7.15 Compatibility between FLACS v10 and v9.1 ................. 22

1.8 FLACS-CFD version history ............................................... 23

1.9 Feedback from users ..................................................... 23

2 Getting started .............................................................. 25
  2.1 Prerequisites for users ................................................ 25
  2.2 Hardware and software requirements .................................. 25
  2.3 Software installation and setup ...................................... 26
    2.3.1 On Linux ............................................................ 27
      2.3.1.1 Installing for a single user ................................ 27
      2.3.1.2 Installing system wide as super user ................. 27
      2.3.1.3 Unattended installation .................................. 27
    2.3.2 On Windows ....................................................... 28
      2.3.2.1 Unattended installation .................................. 28
    2.3.3 Setting up the FLACS-CFD license ......................... 28
      2.3.3.1 License manager runtime software ................. 28
      2.3.3.2 Per user configuration of license .................. 29
      2.3.3.3 Single user license .................................. 29
      2.3.3.4 Network license ..................................... 30
      2.3.3.5 Cloud license (CL) ..................................... 30
        2.3.3.5.1 Availability .................................... 30
        2.3.3.5.2 Technical requirements ......................... 30
        2.3.3.5.3 Troubleshooting the connection to the cloud license server 30
      2.3.3.6 Updating/renewing a license ...................... 30
        2.3.3.6.1 Using the Admin Control Center web interface .... 31
        2.3.3.6.2 Using the FLACS-CFD Configuration Wizard .... 31
        2.3.3.6.3 Using the FLACS-CFD command line license configuration tool 31
      2.3.3.7 Cancelling a license .................................. 31
        2.3.3.7.1 Using the FLACS-CFD Configuration Wizard .... 31
        2.3.3.7.2 Using the FLACS-CFD command line license configuration tool 32
CONTENTS

2.3.8 Checking the number of current sessions ........................................ 33
2.3.9 Common problems when setting up the license ................................. 33
2.4 Setting up the FLACS-CFD environment ............................................ 34
2.4.1 FLACS-CFD User setup on Linux ................................................. 34
2.5 Uninstalling FLACS-CFD ............................................................... 34
2.6 A Tour of FLACS-CFD ................................................................. 34
2.6.1 Things to keep in mind ............................................................. 40
2.6.2 Creating the work directory ....................................................... 40
2.6.3 Starting the preprocessor CASD .................................................. 41
2.6.4 Starting the FLACS-CFD simulation ............................................ 45
2.6.5 Studying the results in Flowvis ................................................... 46
2.6.6 Studying the effect of ignition location ....................................... 49

3 CASD
3.1 Overview ......................................................................................... 51
3.1.1 Geometry in FLACS-CFD ............................................................ 51
3.1.1.1 Geometry representation ....................................................... 52
3.1.1.2 Primitives .............................................................................. 52
3.1.1.2.1 Box ................................................................................. 53
3.1.1.2.2 Cylinder .......................................................................... 54
3.1.1.2.3 Ellipsoid .......................................................................... 54
3.1.1.2.4 General truncated cone (GTC) ........................................ 54
3.1.1.2.5 Convex polyhedron (CP8) .............................................. 54
3.1.1.2.6 Torus ............................................................................... 55
3.1.1.2.7 Rectangular torus (rtorus) .............................................. 55
3.1.1.2.8 Mesh ............................................................................... 55
3.1.1.3 Operations .............................................................................. 55
3.1.1.3.1 Union ............................................................................. 56
3.1.1.3.2 Left difference ................................................................. 56
3.1.1.4 Materials ................................................................................ 56
3.1.1.4.1 UV types ....................................................................... 57
3.1.1.4.2 Drape direction ............................................................... 57
3.1.1.5 Light ...................................................................................... 57
3.1.1.6 Realistic lighting ..................................................................... 58
3.1.1.6.1 Sky light ......................................................................... 58
3.1.1.6.2 Reflections ....................................................................... 58
3.1.1.6.3 Bloom ............................................................................. 59
3.1.2 System requirements ..................................................................... 59
3.1.3 Starting CASD ............................................................................. 60
3.1.4 CASD command line options ....................................................... 60
3.1.5 The main window in CASD .......................................................... 60

FLACS-CFD v22.2 User’s Manual
3.4.1.2.19 Change color ................................................................. 76
3.4.1.2.20 Copy ........................................................................... 76
3.4.1.2.21 Cut ............................................................................... 76
3.4.1.2.22 Paste ............................................................................ 76
3.4.1.2.23 Remove FLACS-CFD geometry ............................................ 76
3.4.1.2.24 Reset FLACS-CFD geometry .................................................. 76
3.4.1.2.25 Replace geometry with FLACS-CFD geometry ......................... 76
3.4.1.2.26 Edit FLACS-CFD geometry .................................................... 76
3.4.1.2.27 Group ............................................................................ 76
3.4.1.2.28 Ungroup ......................................................................... 77
3.4.1.2.29 Create instance... ................................................................. 77
3.4.1.2.30 Replace instance with geometry ............................................... 77
3.4.1.2.31 Edit instance ..................................................................... 77
3.4.1.2.32 Export to terrain ................................................................. 77
3.4.1.3 Add menu ........................................................................... 77
3.4.1.3.1 Box ............................................................................... 77
3.4.1.3.2 Cylinder .......................................................................... 77
3.4.1.3.3 Ellipsoid .......................................................................... 77
3.4.1.3.4 CP8 .................................................................................. 77
3.4.1.3.5 GTC ............................................................................... 77
3.4.1.3.6 Torus ............................................................................... 77
3.4.1.3.7 Rectangular torus ................................................................. 77
3.4.1.3.8 Left difference .................................................................. 78
3.4.1.3.9 Copy ............................................................................... 78
3.4.1.3.10 Object ............................................................................. 78
3.4.1.3.11 From file ......................................................................... 78
3.4.1.4 Select menu ........................................................................ 78
3.4.1.4.1 Bounding box .................................................................... 78
3.4.1.4.2 Filter ............................................................................... 78
3.4.1.4.3 All .................................................................................... 78
3.4.1.4.4 Invert .............................................................................. 78
3.4.1.4.5 Stack ............................................................................... 78
3.4.1.4.6 Group ............................................................................. 79
3.4.1.5 View menu ......................................................................... 79
3.4.1.5.1 The XY, XZ and YZ views ....................................................... 79
3.4.1.5.2 3D View ........................................................................... 79
3.4.1.5.3 Maximize ......................................................................... 79
3.4.1.5.4 Axis ............................................................................... 79
3.4.1.5.5 Terrain ........................................................................... 79
3.4.1.5.6 Grid Display ...................................................................... 79
3.4.1.5.7 Annotation ....................................................................... 79
3.4.1.5.8 Draw Style ....................................................................... 79
3.4.1.5.9 Fixed vertical axis ................................................................. 80
3.4.1.5.10 Fixed near plane ................................................................. 80
3.4.1.5.11 Hide/Unhide ....................................................................... 80
3.4.1.5.12 Flacs view ..................................................................... 80
3.4.1.6 Macro menu .................................................................... 80
3.4.1.6.1 Run ............................................................................... 80
3.4.1.6.2 Record ............................................................................ 80
3.4.1.6.3 Write Object ................................................................. 80
3.4.2 Tool bar .............................................................................. 80
3.4.2.1 Toggle Hide Objects ............................................................ 80
3.4.3 Primitive list ...................................................................... 80
3.4.4 Properties widget ............................................................... 81
<table>
<thead>
<tr>
<th>3.4.5</th>
<th>Quickfilter</th>
<th>81</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4.6</td>
<td>3D view</td>
<td>81</td>
</tr>
<tr>
<td>3.4.6.1</td>
<td>Selecting primitives with the mouse</td>
<td>81</td>
</tr>
<tr>
<td>3.5</td>
<td>Grid menu</td>
<td>82</td>
</tr>
<tr>
<td>3.5.1</td>
<td>Quick grid</td>
<td>82</td>
</tr>
<tr>
<td>3.5.1.1</td>
<td>Core domain</td>
<td>84</td>
</tr>
<tr>
<td>3.5.1.2</td>
<td>Stretched domain</td>
<td>84</td>
</tr>
<tr>
<td>3.5.1.3</td>
<td>Grid information</td>
<td>84</td>
</tr>
<tr>
<td>3.5.2</td>
<td>Check Grid</td>
<td>85</td>
</tr>
<tr>
<td>3.5.3</td>
<td>Simulation volume</td>
<td>86</td>
</tr>
<tr>
<td>3.5.4</td>
<td>Direction</td>
<td>86</td>
</tr>
<tr>
<td>3.5.5</td>
<td>Region</td>
<td>86</td>
</tr>
<tr>
<td>3.5.6</td>
<td>Add</td>
<td>86</td>
</tr>
<tr>
<td>3.5.7</td>
<td>Position</td>
<td>86</td>
</tr>
<tr>
<td>3.5.8</td>
<td>Move</td>
<td>87</td>
</tr>
<tr>
<td>3.5.9</td>
<td>Delete</td>
<td>87</td>
</tr>
<tr>
<td>3.5.10</td>
<td>Smooth</td>
<td>87</td>
</tr>
<tr>
<td>3.5.11</td>
<td>Stretch</td>
<td>87</td>
</tr>
<tr>
<td>3.5.12</td>
<td>Information</td>
<td>87</td>
</tr>
<tr>
<td>3.5.13</td>
<td>List</td>
<td>87</td>
</tr>
<tr>
<td>3.5.14</td>
<td>Display</td>
<td>87</td>
</tr>
<tr>
<td>3.5.15</td>
<td>Select</td>
<td>88</td>
</tr>
<tr>
<td>3.5.16</td>
<td>Grid-related operations</td>
<td>88</td>
</tr>
<tr>
<td>3.5.17</td>
<td>Importing the grid from another job</td>
<td>88</td>
</tr>
<tr>
<td>3.5.18</td>
<td>Saving the grid</td>
<td>88</td>
</tr>
<tr>
<td>3.5.19</td>
<td>Grid-related utilities</td>
<td>88</td>
</tr>
<tr>
<td>3.6</td>
<td>Porosities menu</td>
<td>88</td>
</tr>
<tr>
<td>3.6.1</td>
<td>Calculate</td>
<td>88</td>
</tr>
<tr>
<td>3.6.2</td>
<td>Verify</td>
<td>88</td>
</tr>
<tr>
<td>3.7</td>
<td>Scenario menu</td>
<td>89</td>
</tr>
<tr>
<td>3.7.1</td>
<td>Scenario menu undo/redo</td>
<td>89</td>
</tr>
<tr>
<td>3.7.2</td>
<td>Scenario settings</td>
<td>90</td>
</tr>
<tr>
<td>3.7.2.1</td>
<td>Simulation type</td>
<td>90</td>
</tr>
<tr>
<td>3.7.2.2</td>
<td>Write all variables</td>
<td>90</td>
</tr>
<tr>
<td>3.7.2.3</td>
<td>cs-file compatibility</td>
<td>91</td>
</tr>
<tr>
<td>3.7.2.4</td>
<td>Tags</td>
<td>91</td>
</tr>
<tr>
<td>3.7.2.5</td>
<td>Frequency</td>
<td>91</td>
</tr>
<tr>
<td>3.7.3</td>
<td>Monitor points</td>
<td>91</td>
</tr>
<tr>
<td>3.7.3.1</td>
<td>Naming monitor points</td>
<td>92</td>
</tr>
<tr>
<td>3.7.3.2</td>
<td>Grouping of monitor points</td>
<td>93</td>
</tr>
<tr>
<td>3.7.3.3</td>
<td>Labels for monitor points</td>
<td>93</td>
</tr>
<tr>
<td>3.7.4</td>
<td>Pressure relief panels</td>
<td>94</td>
</tr>
<tr>
<td>3.7.4.1</td>
<td>Name</td>
<td>95</td>
</tr>
<tr>
<td>3.7.4.2</td>
<td>Insert</td>
<td>95</td>
</tr>
<tr>
<td>3.7.4.3</td>
<td>Position</td>
<td>95</td>
</tr>
<tr>
<td>3.7.4.4</td>
<td>Size</td>
<td>95</td>
</tr>
<tr>
<td>3.7.4.5</td>
<td>Material</td>
<td>95</td>
</tr>
<tr>
<td>3.7.4.6</td>
<td>Panel type</td>
<td>95</td>
</tr>
<tr>
<td>3.7.4.6.1</td>
<td>Inactive panel</td>
<td>95</td>
</tr>
<tr>
<td>3.7.4.6.2</td>
<td>Unspecified, Popout and Hinged panel</td>
<td>95</td>
</tr>
<tr>
<td>3.7.4.6.3</td>
<td>Popout rigid and Hinged rigid panel</td>
<td>96</td>
</tr>
<tr>
<td>3.7.4.6.4</td>
<td>Overlay and Plastic panel</td>
<td>96</td>
</tr>
<tr>
<td>3.7.4.6.5</td>
<td>The +IMP panel types</td>
<td>96</td>
</tr>
<tr>
<td>3.7.4.7</td>
<td>Opening pressure differences</td>
<td>97</td>
</tr>
<tr>
<td>3.7.4.8</td>
<td>Initial and final porosity</td>
<td>97</td>
</tr>
<tr>
<td>3.7.4.9</td>
<td>Weight</td>
<td>97</td>
</tr>
<tr>
<td>3.7.4.10</td>
<td>Drag coefficient</td>
<td>98</td>
</tr>
<tr>
<td>3.7.4.11</td>
<td>Maximum travel distance</td>
<td>98</td>
</tr>
<tr>
<td>3.7.4.12</td>
<td>Panel sub-sizes</td>
<td>98</td>
</tr>
<tr>
<td>3.7.4.13</td>
<td>Hinges (flacscfd only)</td>
<td>99</td>
</tr>
<tr>
<td>3.7.4.14</td>
<td>Side outflow (flacscfd only)</td>
<td>99</td>
</tr>
<tr>
<td>3.7.4.15</td>
<td>Using panels in fire simulations</td>
<td>99</td>
</tr>
<tr>
<td>3.7.5</td>
<td>Single field scalar time output</td>
<td>99</td>
</tr>
<tr>
<td>3.7.6</td>
<td>Single field 3D output</td>
<td>100</td>
</tr>
<tr>
<td>3.7.7</td>
<td>Setup</td>
<td>100</td>
</tr>
<tr>
<td>3.7.8</td>
<td>Simulation and output control</td>
<td>100</td>
</tr>
<tr>
<td>3.7.8.1</td>
<td>Start time (TSTART)</td>
<td>101</td>
</tr>
<tr>
<td>3.7.8.2</td>
<td>Minimum time (TMIN)</td>
<td>101</td>
</tr>
<tr>
<td>3.7.8.3</td>
<td>Maximum time (TMAX)</td>
<td>102</td>
</tr>
<tr>
<td>3.7.8.3.1</td>
<td>Example 1: Typical explosion simulation</td>
<td>102</td>
</tr>
<tr>
<td>3.7.8.3.2</td>
<td>Example 2: Explosion simulation with points of interest in the medium/far field</td>
<td>102</td>
</tr>
<tr>
<td>3.7.8.3.3</td>
<td>Example 3: Dispersion or Fire simulation</td>
<td>103</td>
</tr>
<tr>
<td>3.7.8.4</td>
<td>Last timestep (LAST)</td>
<td>103</td>
</tr>
<tr>
<td>3.7.8.5</td>
<td>Load number (LOAD)</td>
<td>103</td>
</tr>
<tr>
<td>3.7.8.5.1</td>
<td>Example 1 - load snapshot #2 into simulation 000000</td>
<td>103</td>
</tr>
<tr>
<td>3.7.8.5.2</td>
<td>Example 2 - load the last available dump snapshot from the CGNS file into simulation 000000</td>
<td>104</td>
</tr>
<tr>
<td>3.7.8.6</td>
<td>Load job name (JLOAD)</td>
<td>104</td>
</tr>
<tr>
<td>3.7.8.6.1</td>
<td>Example 1 - load from specified CGNS file into simulation 000000 and scale if necessary</td>
<td>104</td>
</tr>
<tr>
<td>3.7.8.7</td>
<td>CFLC</td>
<td>105</td>
</tr>
<tr>
<td>3.7.8.8</td>
<td>CFLV</td>
<td>105</td>
</tr>
<tr>
<td>3.7.8.9</td>
<td>Solver mode (MODE)</td>
<td>106</td>
</tr>
<tr>
<td>3.7.8.10</td>
<td>Enable DDT detection (DDT)</td>
<td>106</td>
</tr>
<tr>
<td>3.7.8.11</td>
<td>Force DDT at time (DDT_TIME)</td>
<td>106</td>
</tr>
<tr>
<td>3.7.8.11.1</td>
<td>Example 1: Hydrogen explosion simulation with automatic trigger of detonation</td>
<td>106</td>
</tr>
<tr>
<td>3.7.8.11.2</td>
<td>Example 3: Non-hydrogen explosion simulation with automatic trigger of detonation</td>
<td>107</td>
</tr>
<tr>
<td>3.7.8.11.3</td>
<td>Example 4: Non-hydrogen explosion simulation with specific trigger of detonation</td>
<td>107</td>
</tr>
<tr>
<td>3.7.8.12</td>
<td>Underrelaxation P and UVW (UNDERRELAX_*)</td>
<td>107</td>
</tr>
<tr>
<td>3.7.8.13</td>
<td>Check convergence (CHECK_CONVERGENCE)</td>
<td>107</td>
</tr>
<tr>
<td>3.7.8.14</td>
<td>Convergence tolerance (CONVERGENCE_TOLERANCE_*)</td>
<td>107</td>
</tr>
<tr>
<td>3.7.8.15</td>
<td>Timestep code (STEP)</td>
<td>108</td>
</tr>
<tr>
<td>3.7.8.16</td>
<td>Special control keys (KEYS)</td>
<td>108</td>
</tr>
<tr>
<td>3.7.8.17</td>
<td>MODD</td>
<td>109</td>
</tr>
<tr>
<td>3.7.8.18</td>
<td>N PLOT</td>
<td>109</td>
</tr>
<tr>
<td>3.7.8.19</td>
<td>DT PLOT</td>
<td>109</td>
</tr>
<tr>
<td>3.7.8.20</td>
<td>Incompressible</td>
<td>110</td>
</tr>
<tr>
<td>3.7.8.21</td>
<td>Wall functions (WALLF)</td>
<td>110</td>
</tr>
<tr>
<td>3.7.8.22</td>
<td>Heat switch (HEAT SWITCH)</td>
<td>110</td>
</tr>
<tr>
<td>3.7.8.23</td>
<td>Split CGNS (SPLIT_CGNS)</td>
<td>110</td>
</tr>
<tr>
<td>3.7.8.24</td>
<td>Convergence strategy (CONVERGENCE_STRATEGY)</td>
<td>111</td>
</tr>
<tr>
<td>3.7.9</td>
<td>Time dependent CFL numbers</td>
<td>111</td>
</tr>
<tr>
<td>3.7.10</td>
<td>Dump/load settings</td>
<td>111</td>
</tr>
<tr>
<td>3.7.11</td>
<td>Boundary conditions</td>
<td>112</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>3.7.11.1</td>
<td>Euler</td>
<td>113</td>
</tr>
<tr>
<td>3.7.11.2</td>
<td>Nozzle</td>
<td>113</td>
</tr>
<tr>
<td>3.7.11.3</td>
<td>Plane wave</td>
<td>113</td>
</tr>
<tr>
<td>3.7.11.4</td>
<td>Wind</td>
<td>114</td>
</tr>
<tr>
<td>3.7.11.4.1</td>
<td>Wind wizard</td>
<td>114</td>
</tr>
<tr>
<td>3.7.11.4.2</td>
<td>Wind speed</td>
<td>115</td>
</tr>
<tr>
<td>3.7.11.4.3</td>
<td>Wind direction</td>
<td>116</td>
</tr>
<tr>
<td>3.7.11.4.4</td>
<td>Relative turbulence intensity</td>
<td>116</td>
</tr>
<tr>
<td>3.7.11.4.5</td>
<td>Turbulence length scale</td>
<td>116</td>
</tr>
<tr>
<td>3.7.11.4.6</td>
<td>Wind buildup time</td>
<td>116</td>
</tr>
<tr>
<td>3.7.11.4.7</td>
<td>Best practice</td>
<td>116</td>
</tr>
<tr>
<td>3.7.11.5</td>
<td>Fluctuating wind</td>
<td>117</td>
</tr>
<tr>
<td>3.7.11.6</td>
<td>Using TRACER mode when simulating dispersion</td>
<td>117</td>
</tr>
<tr>
<td>3.7.11.7</td>
<td>Effect of temperature gradients</td>
<td>117</td>
</tr>
<tr>
<td>3.7.11.8</td>
<td>General considerations for boundary conditions</td>
<td>117</td>
</tr>
<tr>
<td>3.7.11.8.1</td>
<td>Solid wall boundary</td>
<td>118</td>
</tr>
<tr>
<td>3.7.11.8.2</td>
<td>External influences</td>
<td>118</td>
</tr>
<tr>
<td>3.7.11.9</td>
<td>Boundary condition at the sea (or other liquid) surface during an explosion</td>
<td>118</td>
</tr>
<tr>
<td>3.7.11.10</td>
<td>Boundary conditions for explosion after dispersion</td>
<td>118</td>
</tr>
<tr>
<td>3.7.12</td>
<td>Initial conditions</td>
<td>118</td>
</tr>
<tr>
<td>3.7.12.1</td>
<td>Up-direction</td>
<td>119</td>
</tr>
<tr>
<td>3.7.12.2</td>
<td>Gravity constant</td>
<td>119</td>
</tr>
<tr>
<td>3.7.12.3</td>
<td>Characteristic velocity</td>
<td>119</td>
</tr>
<tr>
<td>3.7.12.4</td>
<td>Relative turbulence intensity</td>
<td>119</td>
</tr>
<tr>
<td>3.7.12.5</td>
<td>Turbulence length scale</td>
<td>120</td>
</tr>
<tr>
<td>3.7.12.6</td>
<td>Temperature</td>
<td>120</td>
</tr>
<tr>
<td>3.7.12.7</td>
<td>Ambient pressure</td>
<td>120</td>
</tr>
<tr>
<td>3.7.12.8</td>
<td>Air</td>
<td>120</td>
</tr>
<tr>
<td>3.7.12.9</td>
<td>Wind boundary layer parameters</td>
<td>121</td>
</tr>
<tr>
<td>3.7.12.9.1</td>
<td>Ground height</td>
<td>121</td>
</tr>
<tr>
<td>3.7.12.9.2</td>
<td>Pasquill class</td>
<td>121</td>
</tr>
<tr>
<td>3.7.12.9.3</td>
<td>Ground roughness</td>
<td>122</td>
</tr>
<tr>
<td>3.7.12.9.4</td>
<td>Reference height</td>
<td>122</td>
</tr>
<tr>
<td>3.7.12.9.5</td>
<td>Canopy height</td>
<td>122</td>
</tr>
<tr>
<td>3.7.12.9.6</td>
<td>Latitude</td>
<td>122</td>
</tr>
<tr>
<td>3.7.12.9.7</td>
<td>Surface heat P1</td>
<td>122</td>
</tr>
<tr>
<td>3.7.12.9.8</td>
<td>Surface heat P2</td>
<td>122</td>
</tr>
<tr>
<td>3.7.12.9.9</td>
<td>Mean surface heat flux</td>
<td>122</td>
</tr>
<tr>
<td>3.7.13</td>
<td>Gas composition and volume</td>
<td>123</td>
</tr>
<tr>
<td>3.7.13.1</td>
<td>Hydrogen-Nitrogen specific laminar burning velocity model</td>
<td>125</td>
</tr>
<tr>
<td>3.7.13.2</td>
<td>Toxic gas definition</td>
<td>126</td>
</tr>
<tr>
<td>3.7.13.3</td>
<td>Inert gas definition (deprecated)</td>
<td>128</td>
</tr>
<tr>
<td>3.7.13.3.1</td>
<td>Start the Flacs inert simulator</td>
<td>131</td>
</tr>
<tr>
<td>3.7.14</td>
<td>Multiple or custom gas clouds</td>
<td>132</td>
</tr>
<tr>
<td>3.7.14.1</td>
<td>Using predefined cloud shapes</td>
<td>132</td>
</tr>
<tr>
<td>3.7.14.2</td>
<td>Specifying arbitrary cloud shapes</td>
<td>133</td>
</tr>
<tr>
<td>3.7.14.3</td>
<td>Including an existing cloud file</td>
<td>133</td>
</tr>
<tr>
<td>3.7.15</td>
<td>Pool</td>
<td>133</td>
</tr>
<tr>
<td>3.7.15.1</td>
<td>Static pool (STATIC_POOL)</td>
<td>134</td>
</tr>
<tr>
<td>3.7.15.2</td>
<td>Pool shape (POOL SHAPE)</td>
<td>134</td>
</tr>
<tr>
<td>3.7.15.3</td>
<td>Start time (START_POOL)</td>
<td>134</td>
</tr>
<tr>
<td>3.7.15.4</td>
<td>Mass (MASS_POOL_0)</td>
<td>134</td>
</tr>
<tr>
<td>3.7.15.5</td>
<td>DMDT (DMDT)</td>
<td>134</td>
</tr>
<tr>
<td>3.7.15.6</td>
<td>Position (POSITION)</td>
<td>134</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>3.7.15.7</td>
<td>Inner radius and Outer radius (RAD_I and RAD_O)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.8</td>
<td>Ground temperature (T_SOIL)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.9</td>
<td>Heat sun (HEAT_SUN)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.10</td>
<td>Surface roughness (ROUGH_LL)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.11</td>
<td>Ground type (POOL_GROUND)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.12</td>
<td>Ground thermal conductivity (CONDUC_S)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.13</td>
<td>DIFFUS_S (DIFFUS_S)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.14</td>
<td>Air thermal conductivity (CONDUC_A)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.15</td>
<td>Plate thickness (PLATE_L)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.16</td>
<td>Albedo (ALBEDO)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.17</td>
<td>C_POOL (C_POOL)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.18</td>
<td>CFL_POOL (CFL_POOL)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.19</td>
<td>DADT (DADT)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.20</td>
<td>Use database (DATABASE)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.21</td>
<td>EMISSION_A and EMISSION_P (EMISSION_A and EMISSION_P)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.22</td>
<td>H_WATER (H_WATER)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.23</td>
<td>HEAT_EVAP (HEAT_EVAP)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.24</td>
<td>KIN_VISC_A (KIN_VISC_A)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.25</td>
<td>MOL_WEIGHT (MOL_WEIGHT)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.26</td>
<td>PRANDTL_A (PRANDTL_A)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.27</td>
<td>RAD_T (RAD_T)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.28</td>
<td>T_BOIL (T_BOIL)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.29</td>
<td>T_POOL (T_POOL)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.30</td>
<td>Use radiation (POOL_USE_RADIATION)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.31</td>
<td>Fixed Radiation (POOL_FIXED_RADIATION)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.32</td>
<td>Minimum radiation (POOL_MINIMUM_RADIATION)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.33</td>
<td>View factor Six-Flux (POOL_VIEW_FACTOR_6FLUX)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.34</td>
<td>Use turbulence (POOL_USE_TURBULENCE)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.35</td>
<td>Relative turbulence intensity (POOL_RTI)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.36</td>
<td>Turbulence length scale (POOL_TLS)</td>
<td></td>
</tr>
<tr>
<td>3.7.15.37</td>
<td>Leaks</td>
<td></td>
</tr>
<tr>
<td>3.7.16.1</td>
<td>Insert</td>
<td></td>
</tr>
<tr>
<td>3.7.16.2</td>
<td>Type</td>
<td></td>
</tr>
<tr>
<td>3.7.16.2.1</td>
<td>DIFFUSE</td>
<td></td>
</tr>
<tr>
<td>3.7.16.2.2</td>
<td>JET</td>
<td></td>
</tr>
<tr>
<td>3.7.16.2.3</td>
<td>ENTRAINING</td>
<td></td>
</tr>
<tr>
<td>3.7.16.2.4</td>
<td>AIR</td>
<td></td>
</tr>
<tr>
<td>3.7.16.2.5</td>
<td>SUCTION</td>
<td></td>
</tr>
<tr>
<td>3.7.16.2.6</td>
<td>FAN</td>
<td></td>
</tr>
<tr>
<td>3.7.16.2.7</td>
<td>INERT</td>
<td></td>
</tr>
<tr>
<td>3.7.16.3</td>
<td>Leak build-up</td>
<td></td>
</tr>
<tr>
<td>3.7.16.4</td>
<td>Leak concentration between ER0 and ER9</td>
<td></td>
</tr>
<tr>
<td>3.7.16.5</td>
<td>Position</td>
<td></td>
</tr>
<tr>
<td>3.7.16.6</td>
<td>Open sides</td>
<td></td>
</tr>
<tr>
<td>3.7.16.7</td>
<td>Start time</td>
<td></td>
</tr>
<tr>
<td>3.7.16.8</td>
<td>Duration</td>
<td></td>
</tr>
<tr>
<td>3.7.16.9</td>
<td>The Outlet menu</td>
<td></td>
</tr>
<tr>
<td>3.7.16.10</td>
<td>Steps when the jet utility program is not employed</td>
<td></td>
</tr>
<tr>
<td>3.7.16.11</td>
<td>Outlet: area</td>
<td></td>
</tr>
<tr>
<td>3.7.16.12</td>
<td>Outlet: mass flow</td>
<td></td>
</tr>
<tr>
<td>3.7.16.13</td>
<td>Outlet: velocity</td>
<td></td>
</tr>
<tr>
<td>3.7.16.14</td>
<td>Outlet: relative turbulence intensity</td>
<td></td>
</tr>
<tr>
<td>3.7.16.15</td>
<td>Outlet: turbulence length scale</td>
<td></td>
</tr>
<tr>
<td>3.7.16.16</td>
<td>Outlet: temperature</td>
<td></td>
</tr>
</tbody>
</table>
3.7.16.17 Outlet: direction cosines ........................................... 143
3.7.16.18 The Vessel menu ..................................................... 144
3.7.16.19 Fans and suctions .................................................... 144
   3.7.16.19.1 Outlet menu ..................................................... 144
   3.7.16.19.2 Leak file for FAN based on outlet parameters ............ 145
   3.7.16.19.3 Physical modelling of FAN from parameters read from the leak file ......................................................... 145
   3.7.16.19.4 Volume flux due to SUCTION balanced by volume flux entering region ...................................................... 146
   3.7.16.19.5 Verifying the volume flux leaving the computational domain due to SUCTION .................................................. 147
3.7.16.20 Leak Wizard ........................................................... 147
3.7.16.21 Input ........................................................................ 147
3.7.16.22 Output .................................................................... 151
3.7.17 Automatic grid refinement for jet leaks and jet fires ............ 153
3.7.18 Automatic grid refinement for pool fires ............................ 155
3.7.19 Ignition ................................................................. 156
   3.7.19.1 Gas/dust explosion ignition ....................................... 157
   3.7.19.2 Fire Ignition .......................................................... 158
3.7.20 Water spray ............................................................... 158
   3.7.20.1 Insert .................................................................... 159
   3.7.20.2 Position ............................................................... 159
   3.7.20.3 Size ..................................................................... 159
   3.7.20.4 Volume fraction ..................................................... 160
   3.7.20.5 Mean droplet diameter ............................................ 160
   3.7.20.6 Nozzle type .......................................................... 160
3.7.21 Louvre panels .............................................................. 161
   3.7.21.1 Name .................................................................... 164
   3.7.21.2 Position ............................................................... 164
   3.7.21.3 Size ..................................................................... 164
   3.7.21.4 Material .............................................................. 164
   3.7.21.5 Normal vector slats positive ..................................... 164
   3.7.21.6 Normal vector slats negative .................................... 164
   3.7.21.7 Drag coefficient acceleration .................................... 165
   3.7.21.8 Drag coefficient bending ........................................ 165
   3.7.21.9 Drag coefficient friction .......................................... 165
   3.7.21.10 Area porosity ......................................................... 165
3.7.22 Grating ............................................................... 165
   3.7.22.1 Name .................................................................... 167
   3.7.22.2 Position ............................................................... 167
   3.7.22.3 Size ..................................................................... 167
   3.7.22.4 Material .............................................................. 167
   3.7.22.5 Area porosity ......................................................... 168
   3.7.22.6 Characteristic length .............................................. 168
3.7.23 Gas monitor region ......................................................... 168
   3.7.23.1 FlamMass output (FLAMkg in Flacs2) ....................... 168
   3.7.23.2 FlamVol output (FLAM in Flacs2) ............................... 168
   3.7.23.3 GasMass output (TOT_GASkg in Flacs2) .................... 168
   3.7.23.4 ERfacMass output (ERFACkg in Flacs2) ..................... 168
   3.7.23.5 Q6 and Q7 output .................................................. 169
   3.7.23.6 Q8 and Q9 output .................................................. 169
3.7.24 Custom gas monitor regions ........................................ 170
   3.7.24.1 Line monitors ......................................................... 170
   3.7.24.2 Volume monitors .................................................... 171
3.7.25 User species

3.7.25.1 Defining new species in CASD

3.7.25.2 DIPPR

3.7.25.3 Species parameters relevant to all FLACS-CFD applications

3.7.25.4 Species parameters related to combustion

3.7.25.4.1 Fuel composition

3.7.25.4.2 Laminar burning velocity

3.7.25.5 Species parameters related to pool evaporation

3.7.25.5.1 Liquid density

3.7.25.5.2 Specific enthalpy of the liquid phase

3.7.25.5.3 Surface tension

3.7.25.5.4 Vapour pressure

3.7.25.5.5 Liquid viscosity

3.7.25.5.6 Gas viscosity

3.7.25.5.7 Gas thermal conductivity

3.7.25.5.8 Liquid thermal conductivity

3.7.25.5.9 Critical point temperature and pressure

3.7.25.6 Example: definition of a new species, dispersion of fluorine

3.7.26 Radiation

3.7.26.1 Radiation model

3.7.26.2 Emissivity

3.7.26.3 ABSORPTION_COEFFICIENT_MODEL

3.7.26.4 ABSORPTION_COEFFICIENT

3.7.26.5 SCATTERING_COEFFICIENT

3.7.26.6 MEAN_BEAM_LENGTH_MODEL

3.7.26.7 RADIATION_START

3.7.26.8 RADIATION_START_TIME

3.7.26.9 RADIATION_START_RAMP

3.7.26.10 DTM_LOOP_MAX

3.7.26.11 DTM_TOLERANCE

3.7.26.12 DTM_MOD_ITER

3.7.26.13 DTM_MOD_TIME

3.7.26.14 DTM_RAY_DISTRIBUTION

3.7.26.15 DTM_RAYS

3.7.26.16 DTM_VERBOSE

3.7.26.17 DTM_FULL_FLUX

3.7.26.18 DTM_BUFFER_SIZE

3.7.26.19 DTM_FLUX_CUTOFF

3.7.26.20 FAR_FIELD_RADIATION

3.7.26.21 FAR_FIELD_SHADOW

3.7.26.22 DTM_DOMAIN_CONSTRAINTS

3.7.26.23 DTM_DOMAIN_TEMPERATURE_CUTOFF

3.7.26.24 DTM_DOMAIN_POSITION

3.7.26.25 DTM_DOMAIN_SIZE

3.7.27 Combustion

3.7.27.1 COMBUSTION_MODEL

3.7.27.2 EXTINCTION

3.7.27.3 FIRESWITCH

3.7.28 Smoke/soot

3.7.28.1 Soot model

3.7.28.2 Soot yield

3.7.29 Conduction
3.7.30 Solid Flame Model .................................................. 187
3.8 View menu .................................................................... 187
  3.8.1 Print ..................................................................... 187
  3.8.2 Examiner viewer and Fly viewer .............................. 187
  3.8.3 XY View, XZ View, and the YZ Views ...................... 187
  3.8.4 3D View .................................................................. 188
  3.8.5 Axis ..................................................................... 188
  3.8.6 Maximise ................................................................. 188
  3.8.7 Grid display ............................................................. 188
  3.8.8 Annotation ............................................................... 188
  3.8.9 Draw style ............................................................... 188
  3.8.10 LOD and Properties ............................................... 189
  3.8.11 Load filter ............................................................. 189
3.9 Options menu ................................................................. 189
  3.9.1 Units .................................................................... 190
  3.9.2 Preferences ............................................................. 190
    3.9.2.1 General .......................................................... 190
    3.9.2.2 Colours .......................................................... 190
    3.9.2.3 Performance .................................................... 190
  3.9.3 Plugins .................................................................. 191
    3.9.3.1 Creating your own plugins ................................. 191
  3.9.4 Windows ................................................................. 191
  3.9.5 Layout manager ....................................................... 191
3.10 Macro menu (deprecated) .................................................. 191
  3.10.1 Run .................................................................... 192
  3.10.2 Record ................................................................. 192
  3.10.3 Write geometry ..................................................... 192
3.11 Help menu ................................................................. 193
  3.11.1 Online Help ......................................................... 193
  3.11.2 User’s Manual (pdf) ............................................. 193
  3.11.3 Keyboard shortcuts ............................................. 194
  3.11.4 License terms ....................................................... 194
  3.11.5 About CASD ....................................................... 194
3.12 Python API ................................................................. 194
3.13 Tools ................................................................. 194
  3.13.1 Snap points .......................................................... 194
  3.13.2 Terrain import ....................................................... 195
    3.13.2.1 Digital elevation data ...................................... 195
      3.13.2.1.1 Introduction ............................................. 195
      3.13.2.1.2 DEM file formats ..................................... 195
      3.13.2.1.3 Data sets ............................................... 195
    3.13.2.2 Terrain import in CASD .................................... 196
      3.13.2.2.1 Defining a subregion ................................. 197
      3.13.2.2.2 Reducing the terrain resolution ................... 198
    3.13.2.3 Terrain transformations ................................... 199
  3.13.3 Georeferencing ..................................................... 199
  3.13.4 Plan drawings ....................................................... 201
  3.13.5 Clip planes .......................................................... 203
  3.13.6 Arrange items ...................................................... 204
3.14 Working with very large geometries .................................. 205
  3.14.1 Increase the stack size on Linux ............................... 206
  3.14.2 Filter small objects in geo2flacs .............................. 206
  3.14.3 Load filter ........................................................... 206
  3.14.4 Open Inventor viewer enhancements ....................... 206
CONTENTS

3.14.5 Clip planes ................................................. 207
3.15 Known issues in CASD ........................................ 207

4 FLACS-CFD simulators ........................................ 209
4.1 FLACS-Cloud .................................................. 209
4.2 Check list for running simulations ........................... 210
4.3 The RunManager ............................................... 211
  4.3.1 Porosity calculations .................................... 211
  4.3.2 Simulation versions ...................................... 211
  4.3.3 Running jobs locally .................................... 211
  4.3.4 Running multiple simulations in "Batch Run" .......... 212
  4.3.5 Monitoring the simulations ............................. 214
  4.3.6 Setting job parameters ................................ 215
  4.3.7 Suspend, resume and abort simulations .............. 215
  4.3.8 Disk Space usage ....................................... 216
  4.3.9 Automatic halving of CFL numbers .................... 216
  4.3.10 Enabling logging in the RunManager ................. 216
  4.3.11 Known issues with RunManager ....................... 216
  4.4 Managing simulations from the command line .......... 217
    4.4.1 Starting simulations ................................ 217
    4.4.2 Simulator versions ................................... 217
    4.4.3 Monitoring simulations ............................... 217
    4.4.4 Create and use run scripts ......................... 218
    4.4.5 Stopping simulations from the command line ...... 218
    4.4.6 Optimising computer loads ........................ 219
  4.5 Different variants of the simulator ..................... 219
    4.5.1 FLACS-CFD solver licensing ......................... 219
    4.5.2 FLACS-CFD parallel solver .......................... 220
      4.5.2.1 Running the parallel solver from the command line ... 221
      4.5.2.2 Running the parallel solver from the FLACS-CFD RunManager 221
    4.5.3 FLACS-CFD incompressible solver ................... 222
      4.5.3.1 Technical details for the incompressible solver .... 222
      4.5.3.2 Using the incompressible solver ................ 223
      4.5.3.3 Simulating leaks with the incompressible solver .... 223
      4.5.3.4 Time step for the incompressible solver .......... 224
    4.5.4 flacscfd ............................................... 224
      4.5.4.1 Steady-state solver ................................ 225
        4.5.4.1.1 Limitations .................................. 225
        4.5.4.1.2 Stabilising solution ......................... 226
    4.6 Output variables in FLACS-CFD ......................... 226
      4.6.1 General 3D output variables ....................... 230
        4.6.1.1 Mass fraction of fuel: FUEL .................. 230
        4.6.1.2 Pressure: P .................................... 230
        4.6.1.3 Pressure impulse: P_IMP ........................ 230
        4.6.1.4 Mass fraction of combustion products: PROD ........ 230
        4.6.1.5 Gas density: RHO ................................ 231
        4.6.1.6 Gas temperature: T ............................. 231
        4.6.1.7 Turbulence velocity: TURB ..................... 231
        4.6.1.8 Relative turbulence intensity: TURBI .......... 231
        4.6.1.9 Velocity vector: VVEC ......................... 232
        4.6.1.10 Drag value: DRAG ............................. 232
        4.6.1.11 Drag-impulse value: DRAG_IMP ................. 232
        4.6.1.12 Equivalence ratio: ER .......................... 232
        4.6.1.13 Equivalence ratio as %LFL: ERLFL ............. 232
4.6.1.14 Equivalence ratio, normalised DFL: ERNFL .............................. 233
4.6.1.15 Equivalence ratio, finite bounded: EQ .................................... 233
4.6.1.16 Equivalence ratio as %LFL: EQLFL ....................................... 233
4.6.1.17 Equivalence ratio, normalised DFL: EQNFL ............................ 233
4.6.1.18 FMOLE and FDOSE ......................................................... 233

4.6.2 3D output variables for fire simulations .................................... 234
4.6.2.1 Wall temperature: TWALL .................................................. 234
4.6.2.2 Mass fractions: SOOT, H2O, CO2 ........................................ 235
4.6.2.3 Volume/mole fractions: VFSOOT, VFH2O, VFCO2 ...................... 235
4.6.2.4 Visibility ................................................................. 235
4.6.2.5 Convective heat transfer coefficient: HCONV ................................ 235
4.6.2.6 Absorption coefficient: ABSCOF .......................................... 235
4.6.2.7 Radiative source term: RADSRC ........................................... 235
4.6.2.8 Radiative heat flux: QRAD .................................................. 235
4.6.2.9 Far-field radiative heat flux: QRADFF ...................................... 235
4.6.2.10 Convective heat flux: QCONV .............................................. 235
4.6.2.11 Total heat flux: Q ............................................................. 235
4.6.2.12 Wall incident heat flux: QWALL ........................................... 235
4.6.2.13 Incident radiative flux: QRINC ............................................. 235
4.6.2.14 Heat Dose: QDOSE .......................................................... 235

4.6.3 3D output variables for pool simulations .................................... 236
4.6.3.1 Pool depth: POOLD ......................................................... 236
4.6.3.2 Pool volume: POOLVOL ..................................................... 236
4.6.3.3 Pool velocity vector: POOLS ................................................ 236
4.6.3.4 Pool temperature: POOLT .................................................... 236
4.6.3.5 Pool velocity component, x-direction: POOLX ................................ 236
4.6.3.6 Pool velocity component, y-direction: POOLY ................................ 236

4.6.4 General panel output variables .............................................. 237
4.6.4.1 Panel average pressure: PP ............................................... 237
4.6.4.2 Panel average porosity: PPOR .............................................. 237
4.6.4.3 Panel average temperature: PT ............................................ 237
4.6.4.4 Panel average far-field radiation dose: PQFFDOSE ..................... 237
4.6.4.5 Panel average far-field radiative flux: PQRADFF ....................... 237
4.6.4.6 Panel average total heat flux: PQ .......................................... 237
4.6.4.7 Panel average total incident flux: QTINC .................................. 237

4.6.5 Output variables for toxic substances ...................................... 238
4.6.5.1 Toxic mole fraction TMOLE ................................................ 238
4.6.5.2 Toxic dose TDOS .............................................................. 238
4.6.5.3 Toxic probit function PROBIT ............................................. 238
4.6.5.4 Probability of death as function of toxic probit PDEATH .............. 238

4.6.6 Prediction of deflagration to detonation transition using DPDX and DDTLS ........................................ 238

4.6.7 Modifying names and units for output variables .......................... 240

4.7 Files in FLACS-CFD .............................................................. 241
4.7.1 Input files for FLACS-CFD simulations .................................... 242
4.7.1.1 Geometry files ............................................................... 242
4.7.1.2 Grid file ........................................................................ 242
4.7.1.3 Header file ................................................................. 242
4.7.1.4 Porosity file ................................................................. 242
4.7.1.5 Scenario file ............................................................... 242
4.7.1.6 Setup file ................................................................... 242
4.7.1.6.1 The JOBSPEC namelist ................................................ 245
4.7.1.6.2 The SETUP namelist .................................................... 245
4.7.1.6.3 Time stepping for ventilation simulations ......................... 247
4.7.1.6.4 Example using a modified gasdata file for dispersion .......... 247
4.7.1.6.5 Modifying laminar burning velocities in the gasdata file for explosion simulations ......................................................... 249
4.7.1.6.6 The PARAMETERS namelist ......................................................... 249
4.7.1.6.7 Examples of using a setup file to define a high-pressure-temperature region ......................................................... 250
4.7.1.7 Thermal Active Walls ......................................................... 253
4.7.1.8 Cloud file ......................................................... 253
4.7.1.8.1 Potential problems using the cloud interface ......................................................... 256
4.7.1.9 Events file ......................................................... 256
4.7.1.10 Heat file ......................................................... 258
4.7.1.11 Pool setup section ......................................................... 259
4.7.1.12 Pool leakage file ......................................................... 265
4.7.1.13 Leak file ......................................................... 265
  4.7.1.13.1 How to avoid automatic overwriting of leak files by Flacs ......................................................... 267
  4.7.1.13.2 Example 1: Diffuse leak ......................................................... 267
  4.7.1.13.3 Example 2: Jet leak in positive x-direction: ......................................................... 267
  4.7.1.13.4 Example 3: Setting the mixture fraction for leak: ......................................................... 267
  4.7.1.13.5 Example 4: Erroneous specification: ......................................................... 268
  4.7.1.13.6 Example 5: Jet at 45.0 degrees: ......................................................... 268
  4.7.1.13.7 Example 6: Jet at 0.0 degrees: ......................................................... 268
  4.7.1.13.8 Example 7: ......................................................... 268
  4.7.1.13.9 Example 8: ......................................................... 269
4.7.1.14 Runtime simulation control file ......................................................... 269
  4.7.1.14.1 Example 1 ......................................................... 269
4.7.1.15 Time dependent CFL file ......................................................... 270
4.7.1.16 Monitor file ......................................................... 270
  4.7.1.16.1 Line monitor output specification ......................................................... 271
4.7.2 Output files from FLACS-CFD simulations ......................................................... 272
  4.7.2.1 CGNS files ......................................................... 272
    4.7.2.1.1 Writing frequency ......................................................... 274
    4.7.2.1.2 Getting time-series from CGNS file ......................................................... 274
  4.7.2.2 Scalar-time output file (Flacs2 only) ......................................................... 277
  4.7.2.3 Field output file (Flacs2 only) ......................................................... 278
  4.7.2.4 Simulation log file - rt-file ......................................................... 278
    4.7.2.4.1 Volume of calculation domain blocked by geometry ......................................................... 278
    4.7.2.4.2 Flammability limits ......................................................... 279
    4.7.2.4.3 Amount of fuel inside computational domain ......................................................... 279
  4.7.2.5 Simulation log file - tt-file ......................................................... 280
  4.7.2.6 Other simulation log files (Flacs2 only) ......................................................... 280
    4.7.2.6.1 CFL log file ......................................................... 280
    4.7.2.6.2 Fuel log file ......................................................... 281
    4.7.2.6.3 Mass log file ......................................................... 281
    4.7.2.6.4 Pressure log file ......................................................... 282
  4.7.2.7 Simulation dump file (Flacs2 only) ......................................................... 282
  4.7.2.8 Monitor file (Flacs2 only) ......................................................... 282
    4.7.2.8.1 Example: volume with fuel concentration above a defined level ......................................................... 283
  4.7.2.9 Simulation save file ......................................................... 284
4.8 Warnings and error messages ......................................................... 284
  4.8.1 Mass residual ......................................................... 284
  4.8.2 Leak excess area ......................................................... 286
    4.8.2.1 Warnings and error messages for area leaks ......................................................... 286
  4.8.3 CV too small ......................................................... 287
  4.8.4 Errors reading the cs-file ......................................................... 287
  4.8.5 Mixing height <= 0 m ......................................................... 287
# Contents

4.9 Potential bugs and problems with Flacs

4.9.1 Significant overprediction of lower flammability limits for hydrogen+inert mixtures

4.9.2 Underprediction of lift-off length for jet fires

4.9.3 Heavy hydrocarbons C5H12 and upwards

4.9.4 FLACS-CFD underpredicts in empty enclosures

5 Flowvis

5.1 System requirements

5.2 The Flowvis main window

5.2.1 The menu bar

5.2.2 The tool bar

5.2.3 The properties sidebar

5.2.4 The plot area

5.2.5 The time slider

5.2.6 Page selection via tabs

5.3 Plot types

5.3.1 Scalar Time plot

5.3.2 Scalar Line plot

5.3.3 2D cut plane plot

5.3.3.1 Appearance

5.3.3.2 Porosity verification

5.3.4 3D plot

5.3.4.1 Data selection

5.3.4.2 Appearance

5.3.4.2.1 Predefined color setups

5.3.4.2.2 Transparency settings

5.3.4.2.3 Surface appearance options

5.3.4.2.4 2D Slice appearance options

5.3.4.2.5 Glyph appearance options

5.3.4.2.6 Streamline appearance options

5.3.4.2.7 Volume appearance options

5.3.4.3 3D plot options

5.3.4.3.1 Show legend in sidebar

5.3.4.3.2 Grid

5.3.4.3.3 Anti-aliasing

5.3.4.3.4 Ambient occlusion

5.3.4.3.5 Edge enhancement

5.3.4.3.6 Shadows

5.3.4.3.7 Sky

5.3.4.3.8 Show geometry

5.3.4.3.9 Grayscale geometry

5.3.4.3.10 Textured Geometry

5.3.4.3.11 Realistic lighting

5.3.4.3.12 Use Reflections

5.3.4.3.13 Monitor points

5.3.4.3.14 Show Leaks

5.3.4.3.15 Disable Monitor Labels

5.3.4.3.16 Panels

5.3.4.3.17 Set panel transparency from yield

5.3.4.3.18 Show line monitors

5.3.4.3.19 Ignition location

5.3.4.3.20 CAD model

5.3.4.3.21 Domain clip planes

5.3.4.3.22 Picture text box
CONTENTS

5.3.4.3.23 Show title ................................................. 308
5.3.4.3.24 Show description ...................................... 308
5.3.4.3.25 Light position ......................................... 308
5.3.4.3.26 Ground .................................................. 309
5.3.4.4 Visualization of pressure relief panels ................. 309
5.3.4.5 Geometry .................................................. 309
5.3.4.6 Monitor selection ......................................... 310
5.3.4.7 Flight mode ............................................... 311
5.4 Selected topics ............................................... 312
5.4.1 Filter Variables ............................................. 312
5.4.2 Export graphics ............................................. 312
5.4.2.1 Export to GeoTIFF ...................................... 313
5.4.2.2 Batch export ............................................ 314
5.4.3 Export movie ................................................ 314
5.4.4 Export 360 FOV Graphics ................................... 315
5.4.5 Number formats ............................................. 316
5.4.6 Terrain in Flowvis .......................................... 316
5.4.7 Plot linking .................................................. 317
5.4.8 Sidebar legend ............................................... 318
5.4.8.1 Sidebar legend example .................................. 319
5.4.9 Aggregating results ........................................ 320
5.4.9.1 Aggregation dialogue ................................... 321
5.4.9.2 Aggregation example .................................... 322
5.5 Known issues in Flowvis ...................................... 324
5.6 Recommended settings in Flowvis ............................ 325
5.7 Generating a log file in Flowvis ............................. 325
6 Utility programs in FLACS-CFD ................................. 327
6.1 Geometry, grid and porosities ............................... 327
6.1.1 geo2flacs .................................................. 327
6.1.1.1 File formats ............................................. 327
6.1.1.1.1 geo2flacs input file formats ......................... 327
6.1.1.1.2 Output format ....................................... 328
6.1.1.2 Running geo2flacs from CASD ......................... 328
6.1.1.3 Manual geometry import with geo2flacs ............ 328
6.1.1.4 Command line options .................................. 328
6.1.1.5 Geometry verification ................................... 330
6.1.1.6 Importing materials .................................... 330
6.1.1.6.1 Texture filenames ................................... 330
6.1.1.6.2 Texture coordinates for DGN ...................... 330
6.1.1.7 Visualisation and FLACS-CFD model ................ 331
6.1.1.8 Geometry import problems and solutions .......... 331
6.1.1.8.1 Diagnostic output .................................. 331
6.1.1.8.2 Note about PDMS ExPLANT-I ....................... 331
6.1.2 gm ......................................................... 331
6.1.2.1 Examples ............................................... 333
6.1.2.1.1 Create homogeneous grid with 1 m cells .......... 333
6.1.2.1.2 Print grid ........................................... 333
6.1.2.1.3 Smooth grid region .................................. 334
6.1.2.1.4 Refine grid region ................................... 334
6.1.3 FGC (Flacs Geometry Calculator) ........................ 334
6.2 Release source modelling ..................................... 335
6.2.1 Jet .......................................................... 336
6.2.1.1 Grid refinement for time-dependent leaks ........... 340
# CONTENTS

6.2.2 Jet2 .......................... 341
6.2.3 Real gas effects modelled in the jet utilities ............................. 342
6.2.4 Entrainment modelling for leaks ............................... 344
   6.2.4.1 Entrainment functionality ........................................ 344
6.2.5 Grid guidelines for modelled-entrainment leaks ......................... 346
   6.2.5.1 Entrainment model ................................................. 346
   6.2.5.2 Distance to obstacles ............................................ 347
6.2.6 Choosing between jet and modelled-entrainment leaks ................. 347
   6.2.6.1 Best practice .................................................. 348
   6.2.6.2 Particular scenarios in which the entrainment functionality is not applicable 348
6.2.7 Flash .................................. 348
6.3 Modifying simulation files .............................................. 350
   6.3.1 cofile (deprecated) ................................................. 350
6.3.2 coffile2 .............................................. 351
6.3.3 comerge ............................................. 352
6.3.4 autostop ........................................ 353
6.4 Post-processing of legacy simulation output .............................. 354
   6.4.1 r1file ............................................. 354
6.4.2 r3file ........................................... 355
   6.4.2.1 Air change rate .................................................. 357
   6.4.2.2 Creating dump file ................................................. 358
   6.4.2.3 Line monitor output using r3file utility ........................... 358
6.4.3 a1file ............................................. 358
6.4.4 a3file ............................................. 360
   6.4.4.1 Input format for the a3file tool .................................. 361
6.4.4.2 3D output file (r3-file) for a user-defined compound variable ........ 361
6.5 CGNS tools ............................................. 361
   6.5.1 Known issues ................................................. 362
7 FLACS-CFD Best practice .............................................. 363
   7.1 Geometry modelling .............................................. 363
   7.1.1 Creating openings like vents and doors ............................ 363
   7.1.1.1 Option 1 .................................................. 363
   7.1.1.2 Option 2 .................................................. 363
   7.1.2 Left differences closed to the lower boundary ......................... 364
   7.1.3 Advanced left difference operations ................................... 364
   7.2 Terrain roughness ............................................... 365
   7.3 Grid guidelines .............................................. 365
   7.3.1 General guidance .............................................. 365
   7.3.1.1 Grid domains .................................................. 366
   7.3.1.2 Geometry considerations for the grid .................. 367
   7.3.1.2.1 Positioning objects ............................................ 367
   7.3.1.2.2 Pipes .................................................. 367
   7.3.1.2.3 Multiple geometry objects ................................... 367
   7.3.2 Recommended grid configuration ..................................... 368
   7.3.2.1 Grid sensitivity .................................................. 368
   7.3.2.2 Dispersion scenarios .............................................. 368
   7.3.2.3 Using dispersion model output to simulate an explosion ........ 369
   7.3.2.4 Summary of grid recommendations .................................. 370
   7.4 Combined dispersion and explosion simulations with FLACS-CFD .......... 373
   7.5 Simulation example .............................................. 374
   7.5.1 Initialisation .................................................. 374
   7.5.2 Wind and Dispersion Simulations .................................... 374
   7.5.2.1 Scenario .................................................. 374
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.5.2.1.1 Monitor points and output variables</td>
<td>374</td>
</tr>
<tr>
<td>7.5.2.1.2 Simulation and output control</td>
<td>375</td>
</tr>
<tr>
<td>7.5.2.1.3 Boundary Conditions</td>
<td>375</td>
</tr>
<tr>
<td>7.5.2.1.4 Initial conditions</td>
<td>375</td>
</tr>
<tr>
<td>7.5.2.1.5 Gas composition</td>
<td>375</td>
</tr>
<tr>
<td>7.5.2.1.6 Leak</td>
<td>375</td>
</tr>
<tr>
<td>7.5.2.1.7 Ignition</td>
<td>376</td>
</tr>
<tr>
<td>7.5.2.1.8 Gas monitor region</td>
<td>376</td>
</tr>
<tr>
<td>7.5.2.2 Grid</td>
<td>376</td>
</tr>
<tr>
<td>7.5.2.2.1 Quick grid: simulation volume and grid resolution</td>
<td>376</td>
</tr>
<tr>
<td>7.5.2.2.2 Grid refinement</td>
<td>376</td>
</tr>
<tr>
<td>7.5.2.2.3 Manually refine the grid</td>
<td>376</td>
</tr>
<tr>
<td>7.5.2.3 Runtime simulation control file (cc-file)</td>
<td>377</td>
</tr>
<tr>
<td>7.5.2.4 Simulation</td>
<td>378</td>
</tr>
<tr>
<td>7.5.2.5 Results</td>
<td>378</td>
</tr>
<tr>
<td>7.5.3 Explosion simulations</td>
<td>379</td>
</tr>
<tr>
<td>7.5.3.1 Scenario and Grid</td>
<td>379</td>
</tr>
<tr>
<td>7.5.4 Simulation</td>
<td>380</td>
</tr>
<tr>
<td>7.5.5 Results</td>
<td>380</td>
</tr>
<tr>
<td>7.6 Pool spread simulation</td>
<td>381</td>
</tr>
<tr>
<td>7.6.1 Initialisation</td>
<td>381</td>
</tr>
<tr>
<td>7.6.2 Case outline</td>
<td>382</td>
</tr>
<tr>
<td>7.6.3 Grid</td>
<td>382</td>
</tr>
<tr>
<td>7.6.4 Scenario</td>
<td>382</td>
</tr>
<tr>
<td>7.6.4.1 Single field 3D output</td>
<td>383</td>
</tr>
<tr>
<td>7.6.4.2 Simulation and output control</td>
<td>383</td>
</tr>
<tr>
<td>7.6.4.3 Boundary conditions</td>
<td>383</td>
</tr>
<tr>
<td>7.6.4.4 Initial conditions</td>
<td>384</td>
</tr>
<tr>
<td>7.6.4.5 Gas composition and volume</td>
<td>384</td>
</tr>
<tr>
<td>7.6.4.6 Ignition</td>
<td>384</td>
</tr>
<tr>
<td>7.6.4.7 Gas monitor region</td>
<td>385</td>
</tr>
<tr>
<td>7.6.5 Pool setup parameters</td>
<td>385</td>
</tr>
<tr>
<td>7.6.6 Pool leak file</td>
<td>386</td>
</tr>
<tr>
<td>7.6.7 Simulations</td>
<td>387</td>
</tr>
<tr>
<td>7.6.8 Results</td>
<td>387</td>
</tr>
<tr>
<td>7.6.9 Setting the pool position</td>
<td>388</td>
</tr>
<tr>
<td>7.7 How to handle a process stream in FLACS-CFD</td>
<td>391</td>
</tr>
<tr>
<td>7.8 Fire simulation</td>
<td>391</td>
</tr>
<tr>
<td>7.8.1 Initialisation</td>
<td>391</td>
</tr>
<tr>
<td>7.8.2 Grid</td>
<td>392</td>
</tr>
<tr>
<td>7.8.3 Scenario</td>
<td>392</td>
</tr>
<tr>
<td>7.8.3.1 Monitor points</td>
<td>392</td>
</tr>
<tr>
<td>7.8.3.2 Single field scalar time output</td>
<td>392</td>
</tr>
<tr>
<td>7.8.3.3 Single field 3D output</td>
<td>392</td>
</tr>
<tr>
<td>7.8.3.4 Simulation and output control</td>
<td>392</td>
</tr>
<tr>
<td>7.8.3.5 Boundary conditions</td>
<td>393</td>
</tr>
<tr>
<td>7.8.3.6 Initial conditions</td>
<td>393</td>
</tr>
<tr>
<td>7.8.3.7 Gas composition and volume</td>
<td>393</td>
</tr>
<tr>
<td>7.8.3.8 Leaks</td>
<td>393</td>
</tr>
<tr>
<td>7.8.3.9 Ignition</td>
<td>393</td>
</tr>
<tr>
<td>7.8.3.10 Radiation</td>
<td>393</td>
</tr>
<tr>
<td>7.8.3.11 Pool fire</td>
<td>394</td>
</tr>
<tr>
<td>7.8.3.12 Combustion</td>
<td>394</td>
</tr>
<tr>
<td>7.8.3.13 Smoke/Soot</td>
<td>394</td>
</tr>
</tbody>
</table>
7.8.3.14 Conduction .......................................................... 394
7.8.4 Simulations ............................................................... 394
7.8.5 Sample fire simulations ............................................. 394
  7.8.5.1 SINTEF impinging jet fire .................................... 395
  7.8.5.2 U.S. CG China Lake Tests, LNG Pool Fire .................. 399
7.8.6 Source term modelling for pool fires .......................... 401
7.8.7 Coupled pool fires .................................................. 402
7.8.8 Converting an existing dispersion scenario to a jet fire ... 403
7.8.9 Recommended settings in Flowvis for FLACS-Fire .......... 404
7.8.10 FLACS-Fire: Considerations for optimal calculation speed
      7.8.10.1 DTM Ray number ........................................... 404
      7.8.10.2 DTM iteration frequency ................................ 404
      7.8.10.3 DTM Domain ............................................... 404
      7.8.10.4 Incompressible vs. Compressible ....................... 404
7.8.11 Choosing the optimal DTM domain size and far field settings .............. 405
7.9 Equivalent stoichiometric gas cloud ............................ 405
  7.9.1 General Principles ............................................... 405
  7.9.2 Shape of the equivalent gas cloud ............................ 406
  7.9.3 Initial turbulence for the equivalent gas cloud .......... 406
  7.9.4 Ignition of the equivalent gas cloud ....................... 407
  7.9.5 Further guidelines .............................................. 408
7.10 Exporting results for use in structural response software ...... 408
7.11 Setting up a blast simulation with FLACS-Blast ................ 409
  7.11.1 Geometry and grid .............................................. 409
  7.11.2 Monitor points .................................................. 409
  7.11.3 Simulation and output control ............................... 409
  7.11.4 Boundary conditions ......................................... 409
  7.11.5 Explosive charge ............................................... 409
  7.11.6 Example: blast impact on two adjacent structures ....... 410
    7.11.6.1 Problem description ...................................... 410
    7.11.6.2 Computational grid ...................................... 410
    7.11.6.3 Monitor points .......................................... 410
    7.11.6.4 Single field 3D output .................................. 411
    7.11.6.5 Simulation and output control settings ............... 411
    7.11.6.6 Boundary conditions ................................... 411
    7.11.6.7 Initial conditions ...................................... 411
    7.11.6.8 Explosive ................................................ 411
    7.11.6.9 Results .................................................. 411
7.12 Modelling and application limitations .......................... 414
  7.12.1 Geometry modelling ............................................ 414
  7.12.2 Release, dispersion and wind ................................ 415
  7.12.3 Gas explosions ................................................. 417
  7.12.4 Hydrogen explosions ......................................... 418
  7.12.5 Detonation modelling ........................................ 418
  7.12.6 Jet and pool fires ............................................. 419
  7.12.7 Dust explosions ............................................... 419
  7.12.8 Blast wave propagation ...................................... 420
  7.12.9 Terrain representation ...................................... 420
    7.12.9.1 Dispersion ............................................... 420
    7.12.9.2 Explosion and fire ...................................... 421
## 8 Technical Reference

- **8.1 Definitions and gas thermodynamics**
  - 8.1.1 Definitions
  - 8.1.2 Mixing of several gases
  - 8.1.3 Ideal gas relations
    - 8.1.3.1 Isoentropic flow of an ideal gas

- **8.2 Stoichiometric reaction**

- **8.3 Governing equations for fluid flow**

- **8.4 Turbulence modelling**

- **8.5 Wall functions**
  - 8.5.1 Rough wall boundary (flascfd only)

- **8.6 Wind boundary**

- **8.7 Combustion modelling**
  - 8.7.1 The FLACS-CFD flame model

- **8.8 Detonation modelling**

- **8.9 Modelling of jet sources**
  - 8.10.1 Single planar shock model
  - 8.10.2 Ewan–Moodie pseudo-source model

- **8.11 Leaks**
  - 8.11.1 Point leaks
  - 8.11.2 Area leaks
    - 8.11.2.1 Examples of area leaks
  - 8.11.3 Time-varying leaks
  - 8.11.4 Potential issues with oblique jet leaks

- **8.12 Entrainment model**

- **8.13 Jet release of liquefied gas - FLASH utility program**

- **8.14 Numerical schemes**

- **8.15 The frozen cloud concept**
  - 8.15.1 Example: dispersion and ventilation study
  - 8.15.2 Recommendations

- **8.16 Pool model**
  - 8.16.1 Pool motion
  - 8.16.2 Pool heat and mass transfer
  - 8.16.3 Coupled pool fire physics
  - 8.16.4 Humidity and fog
  - 8.16.5 Grid domains for pool scenarios
    - 8.16.5.1 Horizontal simulation domain (X, Y)
    - 8.16.6 Vertical simulation domain (Z)

- **8.17 Fire model**
  - 8.17.1 Combustion Model
    - 8.17.1.1 Eddy dissipation concept
  - 8.17.2 Radiation model
    - 8.17.2.1 Six-flux radiation model
    - 8.17.2.2 Discrete transfer radiation model (DTM)
    - 8.17.2.2.1 Output from the DTM
  - 8.17.2.3 Far-field radiation model
  - 8.17.2.4 Radiative properties model
  - 8.17.3 Soot model
    - 8.17.3.1 Fixed conversion soot model
    - 8.17.3.2 Oxidation Formation Soot Model (FOX)
  - 8.17.4 Conduction
    - 8.17.4.1 Heat equation for walls
    - 8.17.4.2 Wall elements
8.18 Porosity calculations and geometry counting ........................................... 471
  8.18.1 FGC (Flacs Geometry Calculator) .................................................... 472
    8.18.1.1 Gaps in area porosity ................................................................. 472
    8.18.1.2 Trace area porosity ................................................................. 472
    8.18.1.3 Terrain in FGC ................................................................. 472
  8.18.2 cofile ............................................................................................... 473
    8.18.2.1 Counting algorithm in cofile ..................................................... 473
      8.18.2.1.1 Algorithm to count cylinders ............................................... 473
      8.18.2.1.2 Algorithm to count boxes .................................................... 473
      8.18.2.1.3 Counting of objects positioned side-by-side ......................... 474
      8.18.2.1.4 Overlapping cylinders and boxes in the counting algorithms: .... 474
    8.18.2.1.5 Known issues and problems .................................................... 475
  8.18.3 cofile2 ................................................................................................ 475
    8.18.3.1 Known issues and problems with cofile2 ..................................... 475
  8.19 Jet flame shape ....................................................................................... 476
  8.20 Maximum pool area and flame shape for pool scenarios ......................... 477
    8.20.1 Inputs required for flame shape calculation .................................... 477
    8.20.2 Estimating the maximum pool diameter for unconfined pools ........... 478
    8.20.3 Flame shape ..................................................................................... 479
  8.21 The initial high pressure region for a blast ............................................ 482
  8.22 The high pressure region for a blast wave ............................................. 482
  8.23 Linux quick reference ............................................................................ 483
    8.23.1 Distributions .................................................................................... 483
    8.23.2 Desktop environments .................................................................... 483
    8.23.3 Shell .................................................................................................. 484
    8.23.4 Text editors ...................................................................................... 484
    8.23.5 Communication with other computers ............................................ 484
    8.23.6 Help in Linux .................................................................................... 484
    8.23.7 Useful commands ............................................................................. 485
    8.23.8 Permissions ....................................................................................... 486

9 Nomenclature .............................................................................................. 487
  9.1 Roman letters ........................................................................................... 487
  9.2 Greek letters ............................................................................................. 489
  9.3 Subscripts .................................................................................................. 489
  9.4 Dimensionless groups .............................................................................. 490
  9.5 Abbreviations ........................................................................................... 490
  9.6 FLACS-CFD variables ............................................................................. 491

10 Validation .................................................................................................... 493
  10.1 Introduction .............................................................................................. 493
  10.2 Structure of Validation Summary Sheets ............................................... 493
    10.2.1 Experiment details .......................................................................... 493
    10.2.2 FLACS-CFD simulation results ....................................................... 493
      10.2.2.1 Performance metric parameters ................................................. 493
      10.2.2.2 Performance plots ....................................................................... 494
  10.3 Validation cases ........................................................................................ 494
    10.3.1 Discussion .......................................................................................... 494
    10.3.2 Gas Explosion ................................................................................... 497
      10.3.2.1 3D Corner Small Scale .............................................................. 498
      10.3.2.2 Bakerrisk .................................................................................... 500
      10.3.2.3 BFETS Phase 2B ....................................................................... 502
      10.3.2.4 BFETS Phase 3A Deluge ............................................................ 504
      10.3.2.5 BFETS Phase 3A ....................................................................... 506
10.3.2.6 BFETS Partial Fills .......................................................... 508
10.3.2.7 DNVGL 180m3 .............................................................. 510
10.3.2.8 EMERGE ................................................................. 512
10.3.2.9 Fh-ICT Lane Experiment ............................................... 514
10.3.2.10 Fireseal Pipe Penetration Blast Wall ............................. 516
10.3.2.11 FM Global 64m3 empty .............................................. 518
10.3.2.12 FM Global 64m3 Obstacles .......................................... 520
10.3.2.13 FGMGlobal Hydrogen Initial Turbulence ....................... 522
10.3.2.14 FM Global Hydrogen Concentration ............................. 524
10.3.2.15 Hydrogen Balloon .................................................... 526
10.3.2.16 Hydrogen Refueling Station ....................................... 528
10.3.2.17 HySEA Homogeneous ............................................... 530
10.3.2.18 HySEA Uni pi .......................................................... 532
10.3.2.19 JIP MEASURE AR8DR8 SD2 SD9 .................................. 534
10.3.2.20 MERGE ................................................................. 536
10.3.2.21 Mogeleg Branches .................................................... 538
10.3.2.22 MOGELEG Channel .................................................. 540
10.3.2.23 Molkov ................................................................. 542
10.3.2.24 Park: Vented rectangular chamber ............................... 544
10.3.2.25 Shell 30m3 Harrison .................................................. 546
10.3.2.26 Shell Hydrogen Repeated Pipe Congestion .................... 548
10.3.2.27 Shell NaturalHy Repeated Pipe Congestion .................... 550
10.3.2.28 Solvex ................................................................. 552
10.3.2.29 SRI Confined Tube .................................................. 554
10.3.2.30 TNO 39m3 .............................................................. 556
10.3.2.31 Traffic Tunnel Square ............................................... 558

10.3.3 Dispersion ................................................................. 560
10.3.3.1 Burro Tests ............................................................ 561
10.3.3.2 CHRC ................................................................. 563
10.3.3.3 Coyote Tests .......................................................... 565
10.3.3.4 HySEA Inhomogeneous Release .................................. 567
10.3.3.5 LLNL Falcon Tests ................................................... 569
10.3.3.6 Kit Fox ............................................................... 571
10.3.3.7 Maplin Sands ........................................................ 573
10.3.3.8 Thorny Island ......................................................... 575
10.3.3.9 Warehouse Sandia Release ......................................... 577

10.3.4 Fire ................................................................. 579
10.3.4.1 Sandia Cryogenic Hydrogen Jet Fires ............................ 580
10.3.4.2 GL Hydrogen Jet Fire ................................................ 582
10.3.4.3 HSL Hydrogen Impinging Fire ..................................... 584
10.3.4.4 LNG Flare ............................................................ 586
10.3.4.5 NaturalHy Jet Fire .................................................... 588
10.3.4.6 LNG Foamlglas Pool Fire ........................................... 591
10.3.4.7 Montoir LNG Pool Fire ............................................ 593
10.3.4.8 Gasoline 0.3m Pool Fire ........................................... 595

10.3.5 Blast ................................................................. 597
10.3.5.1 Urban Canyon ........................................................ 598

10.3.6 Dust Explosion ........................................................... 600
10.3.6.1 Flame Acceleration Pipe ........................................... 601
10.3.6.2 Silo 12m3 ............................................................. 603
10.3.6.3 Silo 236m3 ............................................................. 605
10.3.6.4 Vented Duct 18.5m3 ................................................ 607
<table>
<thead>
<tr>
<th>11 References</th>
<th>609</th>
</tr>
</thead>
<tbody>
<tr>
<td>11.1 Bibliography</td>
<td>609</td>
</tr>
<tr>
<td>11.2 Online references</td>
<td>617</td>
</tr>
<tr>
<td>Index</td>
<td>619</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 About this publication

FLACS-CFD 22.2 User Manual

Copyright © 2023 Gexcon AS
All rights reserved
Updated: February 20 2023
Printed in Norway

Intellectual property notice
No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without written permission from Gexcon AS.

Gexcon AS hereby grants permission to use, copy, and print this publication to organisations or individuals holding a valid license for one or several of the software packages described herein.

For further information about Gexcon AS, please visit the Gexcon web site.

Exclusion of liability
Gexcon AS has distributed this publication in the hope that it will be useful, but without any warranty, without even the implied warranty of merchantability or fitness for a particular purpose.
Introduction

Although great care has been taken in the production of this publication to ensure accuracy, Gexcon AS cannot under any circumstances accept responsibility for errors, omissions, or advice given herein.

Registered trademarks

- FLACS-CFD, FLACS, CASD, and Flowvis are registered trademarks of Gexcon AS.
- Linux is a registered trademark of Linus Torvalds.
- Windows is a registered trademark of Microsoft Corporation.

Other product names mentioned herein are used for identification purposes only and may be trademarks of their respective companies.

1.2 Preface

Computational fluid dynamics (CFD) is a branch of fluid mechanics that uses numerical methods and algorithms to solve and analyse problems that involve fluid flow, with or without chemical reactions. Current use of CFD covers a broad range of applications, from fundamental theoretical studies involving models primarily derived from first principles, to practical engineering calculations utilising phenomenological or empirical correlations.

Many of the hazards encountered in society, and especially in the process industries, involve accident scenarios where fluid flow in complex, large-scale, three-dimensional (3D) geometries play a key role. FLACS-CFD is a specialised CFD toolbox developed especially to address process safety applications such as:

- Dispersion of flammable or toxic gas
- Gas and dust explosions
- Propagation of blast and shock waves
- Pool and jet fires.

The development of FLACS-CFD started in 1980 at the Department of Science and Technology at Christian Michelsen Institute (CMI). CMI established Gexcon (Global Explosion Consultants) as a consultancy activity under the Process Safety Group in 1987. In 1992, the Science and Technology department at CMI became Christian Michelsen Research (CMR), and CMR established Gexcon as a private limited company in 1998.

The purpose of this manual is primarily to assist FLACS-CFD users in their practical work with the software. In addition, the manual aims at documenting both the physical and chemical models, and the numerical schemes and solvers, implemented in the CFD code. Ample references to published literature describe the capabilities and inherent limitations of the software.

1.3 Application areas

FLACS-CFD is a specialised computational fluid dynamics (CFD) tool for safety applications. Major disasters continue to cause severe losses in industry and society in general, and some of the most severe accident scenarios entail fluid flow, with or without chemical reactions, as well as blast wave propagation, in complex geometries. Examples of such events include (Mannan, 2012):

- Loss-of-containment (release) and dispersion of flammable, asphyxiating, malodorous, toxic and/or radioactive material in gaseous, liquid and/or solid form;
- Gas explosions, vapour cloud explosions (VCEs), mist explosions, dust explosions, colliery explosions, hybrid explosions and vapour explosions (physical explosions);
1.3 Application areas

- Detonation of condensed explosives and propagation of blast waves (accidents or malicious attacks);
  and

- Jet fires and pool fires.

A majority of the 100 largest property losses in the hydrocarbon industries from 1972 to 2011 involved fires and explosions (Marsh, 2012).

In their daily work, safety engineers use a variety of tools for assessing the consequences of accident scenarios, and for optimising the design of process facilities and safety measures. Examples of such models include:

- Simple analytical expressions and empirical correlations, typically prescribed in safety standards and implemented in software packages,

- Phenomenological tools of varying complexity,

- Sophisticated CFD tools that account for the actual initial and boundary conditions, and solve the governing equations for conservation of mass, momentum and energy.

Computational fluid dynamics (CFD) represents the state-of-the-art in consequence assessment for flow-related accident scenarios. In some situations, the simpler models may under- or overpredict the consequences by a factor ten or more (Zalosh, 2008), even within their prescribed range of application. Regardless of the complexity of the model used, it is essential for the quality of a quantitative risk assessment (QRA), and hence for the safety and security in the facility, that safety engineers understand the underlying assumptions and inherent limitations of the tools they use, as well as the level of accuracy they can expect in the results. New users of FLACS-CFD must attend an introductory training course that conveys adequate knowledge of fluid dynamics and teaches the proper interpretation of simulation results. Beyond that, this User Manual provides comprehensive information on how to set up simulations, the modelling and theory underlying FLACS-CFD, and its limitations.

The following paragraphs summarise the main application areas of the CFD tool FLACS-CFD, and highlights key aspects of the model system.

1.3.1 Computational fluid dynamics

Although the governing equations for turbulent fluid flow are well established (Bradshaw, 1994), analytical solutions are primarily of academic interest, and discrete solutions by direct numerical simulation (DNS) can only be realized for idealised systems. In recent years, models based on large eddy simulations (LES) have gained increasing popularity at universities. However, within the context of simulating industrial accident scenarios, most commercial CFD tools, including FLACS-CFD, still rely on turbulence models based on Reynolds-averaged Navier-Stokes (RANS) equations, such as the $k - \epsilon$ model (Launder & Spalding, 1974), complemented with sub-grid models to account for the influence of objects that cannot be resolved on the computational grid. For turbulent reactive flows, it is necessary to add models for chemical reactions, and to couple the resulting model system (Hjertager, 1982). When it comes to describing real industrial systems, it is important for users of advanced CFD tools to keep in mind that most simulations are inherently ‘under-resolved’, and that a significant degree of sub-grid modelling is required. This implies that solutions may not converge as the spatial or temporal resolution increases, and it is important to follow the guidelines provided by the software vendor.

1.3.2 The model system

FLACS-CFD is an advanced tool for performing engineering calculations related to safety and security in industry and society in general. The software includes a 3-dimensional (3D) CFD code that solves Favre-averaged transport equations for mass, momentum, enthalpy ($h$), turbulent kinetic energy ($k$), rate of dissipation of turbulent kinetic energy ($\epsilon$), mass-fraction of fuel ($Y_F$) and mixture-fraction ($\xi$) on a structured Cartesian grid using a finite volume method. The RANS equations are closed by invoking the ideal gas equation of state and the standard $k - \epsilon$ model for turbulence (Launder & Spalding, 1974).
FLACS-CFD solves for the velocity components on a staggered grid, and for scalar variables, such as density, pressure and temperature, on a cell-centred grid. The accuracy of the Flacs solver is second order in space and first/second order in time. FLACS-CFD uses the SIMPLE pressure correction scheme (Patankar, 1980) extended with source terms for the compression work in the enthalpy equation for compressible flows, and the SIMPLEX scheme for non-compressible flows.

One of the key features that distinguishes FLACS-CFD from most commercial CFD codes is the use of the distributed porosity concept for representing complex geometries on relatively coarse computational meshes. With this approach, large objects and walls are represented on-grid, whereas smaller objects are represented sub-grid. The pre-processor FGC reads the grid and geometry files and assigns volume and area porosities to each rectangular grid cell. In the simulations, the porosity field represents the local congestion and confinement, and this allows sub-grid objects to contribute with flow resistance (drag), turbulence generation and flame folding in the simulations.

The use of CFD for consequence assessments is not limited to any particular market or industry. The basic conservation laws apply equally well to safe design and optimisation of new technology, including the emerging field of renewable energy: the use of hydrogen as an energy carrier, pipeline transportation of carbon dioxide as part of the carbon capture and storage (CCS) chain, production and use of various types of biofuel, etc. There is significant potential for increased use of CFD with respect to simulating accident scenarios in industry and society in general.

1.3.3 Release and dispersion

FLACS-CFD models flow in the atmospheric boundary layer (wind) by forcing profiles for velocity (wind speed and wind direction), temperature and turbulence parameters ($k$ and $\epsilon$) at the inlet boundaries. The choice of profile should reflect the surface roughness and atmospheric stability class (Pasquill class) or the Monin-Obukhov length. The model accounts for buoyancy effects through additional terms in the momentum and turbulence model equations.

1.3.4 Gas explosions

While premixed combustion under constant volume or constant pressure conditions is straightforward to describe, gas explosions in complex industrial environments are complex phenomena. The key physical phenomenon to model is the positive feedback loop between expansion-generated flow and increased rate of turbulent combustion, which leads to flame acceleration, pressure build-up and generation of blast waves. A suitable starting point for the novice in the field of gas explosions is the Gas Explosion Handbook (Bjerketvedt et al., 1997).

The development of dedicated CFD codes for simulating gas explosions in realistic industrial geometries started around 1980. The combustion models implemented in explosion codes for industrial applications usually assume premixed combustion and fast chemistry. The main source of validation data are extensive series of large-scale experiments with natural gas, and the main area of application is offshore installations. The significant spread in the results from repeated explosion experiments at large scales represents a challenge for model validation (Evans et al., 1999; Skjold et al., 2013).

The current trend is to apply the same CFD tools to risk assessments for onshore processing facilities. This creates several challenges with respect to model validation. There is increasing awareness of the potential for realising deflagration-to-detonation transition (DDT) during flame propagation in large congested fuel-air clouds (HSE, 2009; Johnson, 2010; Tomlin & Johnson, 2013). Flammable mixtures containing gaseous fuels such as propane or ethylene react far more violently than methane-air mixtures. The research community has limited understanding of the effects of flow-turbulence interactions, anisotropic effects, and the influence of Landau-Darrieus, Kelvin-Helmholtz, Rayleigh-Taylor and/or Richtmyer-Meshkov instabilities on flame speed and pressure build-up, especially in large-scale industrial geometries. Hence, Gexcon invests significant resources in developing improved combustion and turbulence models that can be applied on a coarse computational mesh, and still capture the dominating mechanisms for flame propagation. The technical reference chapter describes the combustion model used in FLACS-GasEx.
1.3 Application areas

1.3.5 Hydrogen safety

FLACS-Hydrogen is a special variant of FLACS-CFD developed for hydrogen safety. The functionality is similar to FLACS-GasEx and FLACS-Dispersion, but limited to hydrogen as fuel.

1.3.6 Dust explosions

FLACS-DustEx, previously known as DESC (Dust Explosion Simulation Code), is a special variant of FLACS-CFD developed for simulating industrial dust explosions. Users define a dedicated combustion model for a specific dust from pressure-time data experimentally determined in a standard 20-litre explosion vessel (Skjold, 2007). FLACS-DustEx treats the dust cloud as a dense gas, assuming that the dispersed phase (particles) is in thermal and kinetic equilibrium with the continuous phase (typically air).

The specifics of FLACS-DustEx are not covered in this manual.

1.3.7 Blast wave propagation

FLACS-Blast (previously known as FLACS-Explo) is a special variant of FLACS-CFD that simulates the propagation of blast waves arising from the detonation of condensed explosives. The Blast simulator does not model the detonation process itself but a specified amount of explosive material is transformed into a high-pressure high-temperature region that is used as initial condition for the simulation. FLACS-Blast solves the Euler equations with a conservative shock-capturing scheme, the so-called flux-corrected transport (FCT) scheme (Boris & Book, 1973), together with the SOLA-ICE algorithm and a second order flux correction, instead of the SIMPLE algorithm used by the standard FLACS-CFD gas explosion simulator. Since the Euler equations are solved, it is not required to include sub-grid contributions such as turbulence. The porosity file is binarised (grid cells with less than 50% porosity are assumed to be fully blocked, the rest are fully open). FLACS-Blast treats the explosive like a bursting “balloon”, which starts with the initial condition of a sphere of high temperature and pressure gases at zero velocity. The balloon analogue method has been described previously (Brode, 1959; Ritzel and Matthews, 1997; Donahue, et al., 2004). The equation of state for the ideal gas law (ratio of specific heats equal to 1.4) is used, which corresponds to air (or any other diatomic gas). The scaling relation for the source volume is obtained from the standard TNT blast curves, so that the diameter of the “balloon” is proportional to the mass of explosive material. The initial condition is calculated based on the heat of reaction and the temperature of the combustion products (for TNT detonations), which are then isentropically compressed to a pressure of 808 bar and a temperature of around 10,000 K (consistent with the TNT blast curve). FLACS-Blast also has relationships for RDX, where the isentropically compressed state of the “balloon” is 936 bar and 13,288.6°C. Since there is no case-specific calibration involved, it is possible to perform reasonably accurate predictions within the inherent limitation of the achievable grid resolution (Nolde & Skjold, 2010; Skjold et al., 2012; Davis & Hinze, 2014).

The Best practice chapter contains an example of how to set up a blast simulation, and the Modelling and application limitations section details some limitations of FLACS-Blast.

1.3.8 Future developments

Gexcon has plans to further develop FLACS-CFD so that it can simulate mist and spray combustion (explosions), as well as hybrid explosions. The steady increase in computational resources (speed, memory, etc.), accompanied by parallelisation of the numerical solvers, allows for faster calculations on larger computational domains. However, there is still a need for further speed-up, improved accuracy, and reduced sensitivity of the results with respect to the spatial resolution used in the simulations; technology based on adaptive mesh refinement (AMR) and hybrid parallelisation represents a promising solution for achieving this goal.
1.3.9 Validation and documentation

The validation and documentation process represents a fundamental challenge for developers of any model system that aspire to describe a wider range of physical phenomena, or other initial and boundary conditions than the ones that can be mapped out by a finite number of experiments. Both government bodies and industry show increasing awareness of the need to qualify models for particular applications, for instance by requiring modellers to demonstrate the capabilities of their models to reproduce results from specific sets of experiments (Ivings et al., 2007). The current trend in software development for application-specific CFD tools, such as FLACS-CFD, entails an integrated framework for model validation, implemented as a natural extension of the continuous integration and life-cycle management system for the software (Skjold et al., 2013).

1.4 Acknowledgements

The development of FLACS-CFD would not have been possible without the generous contributions received from supporting companies and government institutions throughout the years. The activity started at Christian Michelsen Institute (CMI) in 1980 with the Gas Explosion Programs (GEPs), and FLACS-86 was the first version distributed to the supporting companies.

The development of FLACS continued with the Gas Safety Programs (GSPs) and related projects up to around 2000:

- BP, Elf, Esso (Exxon), Mobil, Norsk Hydro, and Statoil supported the development of FLACS-86 during the First GEP (1980-1986).
- BP, Mobil, and Statoil supported the development of FLACS-89 during the Second GEP (1986-1989).
- BP, Elf, Esso, Mobil, Norsk Hydro, Statoil, Conoco, Philips Petroleum, Gaz de France, NV Nederlandse Gasunie, Bundes Ministerium für Forschung und Technologie (BMFT), Health and Safety Executive (HSE), and the Norwegian Petroleum Directorate supported the development of FLACS-93 during the First GSP (1990-1992).
1.4 Acknowledgements


- BP, TotalElfFina (TEF), Norsk Hydro, Statoil, Gaz de France, Philips Petroleum, Mobil and supported the LICOREFLA project (2000-2001).

Since 2000, the more recent FLACS releases have been supported by various Joint Industry Projects (JIPs), funding from the European Commission (EU), the Norwegian Research Council (NFR), and support and maintenance fees from an increasing number of commercial customers. In addition, several specialised versions of FLACS-CFD, such as DESC (Dust Explosion Simulation Code), FLACS-Dispersion, and FLACS-Hydrogen have been developed. The chronological development has been:

- FLACS-Dispersion and FLACS-Hydrogen became available in 2001 (hydrogen had been available as a gas in FLACS since 1989, but can be purchased as a dedicated tool since 2001. FLACS-Hydrogen was strongly improved with FLACS 8.1 in 2005.).

- FLACS v8.0 was released in 2003, including a test release of FLACS-Explo.

- FLACS v8.1 was released in 2005.

- DESC 1.0 was released in 2006.

- FLACS v9.0 was released in 2008, including a test release of FLACS-Fire.

- FLACS v9.1 was released in November 2009.

- FLACS v10.0 was released in December 2012.

- FLACS v10.1 was released in May 2013.

- FLACS v10.2 was released in December 2013.

- FLACS v10.3 was released in July 2014.

- FLACS v10.4 was released in June 2015.

- FLACS v10.5 was released in May 2016.

- FLACS v10.6 was released in April 2017.

- FLACS v10.7 was released in November 2017.

- FLACS v10.8 was released in summer 2018.

- FLACS v10.9 was released in summer 2019.

Gexcon has also developed several in-house R&D tools, including FLACS-Aerosol and FLACS-Energy.

Gexcon is grateful to all companies, government institutions, and individuals that have participated in the development of FLACS-CFD. We intend to honour these contributions by continuing to develop the software, and thereby contribute to improved safety in the process industries.
1.5 About this manual

This User Manual describes a family of computational fluid dynamics (CFD) software products from Gexcon AS, generally referred to as FLACS-CFD:

- The preprocessor CASD
- The CFD simulator Flacs
- The postprocessor Flowvis
- Utility programs in FLACS-CFD such as:
  - geo2flacs, gm, and FGC (Flacs Geometry Calculator)
  - jet and flash
  - rdfile, cofile (deprecated), and comerge
  - r1file, r3file, a1file, and a3file

These programs constitute a specialised CFD tool, FLACS-CFD, or ‘standard FLACS-CFD’, designed to study releases of flammable gas and gas explosions in complex congested geometries, both onshore and offshore.

Note:

A newer version of this User Manual might be available on the FLACS-CFD User Portal.

Note:

In some cases, the cached help file may be out of sync with the installed FLACS-CFD manual. If an older version of the manual opens when a newer version is expected, then the cache must be deleted. This can be achieved by removing the following directory with all of its contents:

Windows: C:\Documents and Settings\USER_NAME\Local Settings\...
... Application Data\assistant\gexcon\flacs-cfd_22_2
or: C:\Users\USER_NAME\AppData\Local\assistant\gexcon\flacs-cfd_22_2

Linux: ~/.local/share/data/assistant/gexcon/flacs-cfd_22_2
or ~/.local/share/gexcon/flacs-cfd_22_2

The acronym FLACS-CFD (FLame ACceleration Simulator) refers to the complete package of products, whereas the term flacs and flacs2 refers specifically to the numerical solver in the CFD code.

FLACS version 9.0 (FLACS v9.0) represented a major upgrade to the graphical user interfaces (GUIs), and was the first version that ran under both the Linux and Windows operating systems.

Getting started presents a detailed example for new users of FLACS-CFD, and Best practice examples contains further examples that highlight various applications of FLACS-CFD, including some of the specialised variants.

Technical reference contains technical reference material.

1.5.1 Printing conventions in this manual

- The symbol ‘>’ followed by text in typewriter font indicates command line input, e.g.:

```bash
> command -options arguments  (general syntax for commands)
> find -name flacs           (command line input in Linux)
```
1.5 About this manual

- The symbol ‘∗’ followed by text in typewriter font field input commands, e.g.:
  ∗ exit yes yes
- The symbol → indicates a path through nested menu items or dialog box options, e.g.:
  File → Save
  Scenario → Ignition → Time of ignition
- Certain features of the software may only be accessible through text file input, and the content of a text file is also printed in typewriter font:
  
  THE FIRST LINE OF THE FILE ...
  THE SECOND LINE OF THE FILE ...
  ...
  ...
- The format for describing keyboard and mouse input follows the pattern:
  CTRL+C
  CTRL+left_click
- The use of **bold** or *italic* font emphasises specific words or phrases in the text.
- The Nomenclature chapter lists the symbols and abbreviations adopted in this manual.

1.5.2 Special messages

**Warning:**

Look out for the potential pitfalls pointed out by this heading!

**Note:**

Be aware of practical information pointed out by this heading.

**Remarks:**

Take notice of the points summarised under this heading.

**See also:**

Follow up the additional sources of information suggested by this heading if required.

1.5.3 Job numbers

The typical application of FLACS-CFD is to quantify potential consequences of industrial accident scenarios involving compressible fluid flow, with or without chemical reactions. Proper characterisation of a particular problem may involve several simulations, and it is usually convenient to organise the files from related scenarios in a dedicated directory. The individual FLACS-CFD simulations are assigned job numbers, or simulation numbers, or simply jobs. In Linux, to start job number 010100 type on the command line:

```
> run flacs 010100
```

The job numbers are constructed from a six-digit string \(ijklmn\), where traditionally:

- \(ij\) is the project number.
- \(kl\) is the geometry number.
- \(mn\) is the sequence number.

The default job number used in many of the examples in this manual is 010100, i.e. project 01, geometry 01, simulation 00. However, each of the six digits in the job number may in principle take on any integer value from zero to nine, and the references to project, geometry, and sequence numbers only apply when the job numbers are derived from the file database in CASD.
1.6 What is new in this release?

1.6.1 FLACS-CFD 22.2

1.6.1.1 Simulators

- Added detonation model, activated by default for hydrogen. For details refer to section Detonation modelling and Validation.

- Corrected expansion ratio in update routine leading to marginal change of simulated pressure in explosion scenarios.

- Automatic rewind and restart of the simulation with reduced time step in case of convergence errors ('LARGE MASS RESIDUAL' condition).

- Splitting of the CGNS file.

- Fixed bug in FLACS-DustEx parallel solver resulting in random unphysically high pressure peaks.

- Fixed issue affecting some integral variables related to the toxicity and heat dose resetting to 0 after resuming or loading from the dump snapshot.

- Switched to using Le Chatelier's mixing rule to calculate LFL for mixtures.

- Corrected UFL for mixtures, removing artificial tail with zero values from the laminar burning velocity curve.

- Corrected bug in HINGED panels, fixed missing conversion from radians to degrees. The bug affected X-direction and Y-direction (not Z-direction) HINGED panels resulting in a slower opening of the panel and thereby causing higher overpressure (in most cases).

- Fixed bug causing panels placed on any of the upper boundaries of the domain to disappear.

- Fixed bug in diffuse-type area leaks causing incorrect leak dimension and reduced mass rate (affected only releases in directions X and Y).

- Fixed bug in area leaks with mixed stream (option :mix) causing incorrect fuel rate.

1.6.1.2 FGC

- Improved handling of mesh geometry and rotated primitives.

- Added a new option named "Trace area porosity" to avoid gaps and to ensure water tight geometry.

1.6.1.3 CASD

- The geometry database concept has been removed.

- The coXXXXXX.dat3 file is no longer generated when saving the scenario.

- The FLACS model is no longer generated when importing CAD geometry or when adding mesh primitives (torus and rectangular torus).

- Improved handling of mesh geometry in the object editor.

- Added functionality in the object editor for slicing geometry.

- Menu choices for importing CAD, terrain, and geometry files have been moved to File->Import in the object editor.

- Added option in the object editor to override the "Trace area porosity" setting on individual primitives.
1.6 What is new in this release?

- Added support for instanced geometries in coXXXXXX.geo file.
- Terrains are now stored in the coXXXXXX.geo file and no longer in the .tri file.
- Multiple terrains can now be added to the geometry.
- The "Edit terrain..." menu choice has been removed and the terrains can be transformed the same way as other primitives.
- Changed default behavior in Volume Fraction dialog to only show a subset of all the species.
- Renamed variable names in scenario menu to be same as output variables in CGNS file.
- Added a new item in the scenario item menu for controlling the splitting of CGNS result files.
- Fixed issue with dragging bounding box not behaving as expected.
- Fixed issue with importing FLACS modules in Python console window.
- Warn user when toxic specification is left blank.
- Added constraints to GUI so that negative values are not accepted for parameters that should only be positive.
- Fixed issue that was preventing change of HUE on panels.
- Fixed issue with copy/pasting groups of panels or monitor points.
- Fixed issue in grouping functionality for panels and monitor points.

1.6.1.4 Flowvis

- Fixed holes in ISO-surfaces.
- Fixed issue with textured geometry not visible when opened in Flowvis.
- Fixed issue with 1D variables available for selection in 3D plots.
- Fixed issue with porosity verification not possible without CS file.
- Fixed issue with automatic simulation range for VelocityVector 3D not working.
- Fixed issue in 2D Cut Plane with domain variables values provided in the X Y Z fields changes when switching the domain option.

1.6.1.5 RunManager

- Actions like "Simulate" and "Abort simulation" are now performed on checked/ticked jobs and no longer on the current selected/highlighted job. This makes it possible to start multiple simulations at the same time without using the Batch functionality.
- Warning dialogs have been added to prevent unwanted overwriting of data when restarting simulations that are suspended or finished.
- Added button to upload scenario files to FLACS-Cloud.
- A new status named "BLOCKED" has been added, indicating that a CGNS file is open in another application, blocking the simulator from opening and writing to the file.
- Improved performance when uploading files to FLACS-Cloud.
- Improved stability when connected to FLACS-Cloud.
1.6.2 FLACS-CFD 22.1r2

1.6.2.1 CASD

- Fixed issue related to pool fire scenario failing in solver (gives error about obsolete CS.POOL file).
- Fixed issue related to volume fraction for user species 2 & 3 being reset to zero when loading scenario.
- Fixed issue related to VISIB output variable not being available for fire simulations.
- Fixed issue related to leak wizard not working for n-BUTANE.
- Fixed issue related to the results not reflecting the CD value entered in the jet leak wizard when typed in instead of selected from the drop down.
- Fixed issue related to the scrollbar in the information box in Check Grid dialog being disabled.

1.6.2.2 Flowvis

- Fixed issue related to monitor points not being correctly sorted in ascending order.

1.6.3 FLACS-CFD 22.1

1.6.3.1 Simulators

- Improved flame radius model.
- Improved grid sensitivity.
- Corrected wall shear stress source terms.
- Improved model of shear stress in the turbulence production term.
- Solution is now automatically dumped when maximum time and at fuel level stop criteria are reached.
- Fixed setting thermal properties for pool substrate.
- Removed message '*** ERROR: Mixing height <= 0 m' and allowed zero velocity at wind boundaries.
- Fixed reading of cp-files larger than 2 GB.
- Fixed reading of species names that contain special characters.
- Fixed toxic properties file clashing with multiple leak files.

1.6.3.2 CASD

- Added grid check tool to help evaluate grid quality for current scenario against grid guidelines
- Allowing to close object viewer with empty primitive list.
- Removed the "Remember to calculate porosities..." dialog that appear on every save.
- Fixed issue in grid refinement tool for jet leaks related to updating parameter values.
- Fixed issue related to case conversion of ":mix" string in leak description in scenario file.
- Fixed issue related to grid changes when switching units in Options.
- Fixed issue related to opening a new scenario in CASD when a modified scenario is already open.
- Fixed issue related to Cancel not working properly in geometry counting dialog.
- Fixed possible crash on exit.
1.6.3.3 Flowvis

- Fixed issue related to using non default units and switching jobs.
- Fixed issue related to geometry colormap not properly restored from presentation.

1.6.3.4 RunManager

- Fixed bug leading to crash when connected to FLACS Cloud.
- Fixed bug related to deletion of remote files when using FLACS Cloud.
- Fixed enabling of Abort and Suspend/Stop buttons when running porosity calculations.

1.6.4 FLACS-CFD 21.3

1.6.4.1 Simulators

- Parallelisation and performance optimisation of the DTM.
- Performance and memory optimization of FLACS-Fire.
- Added uncoupled pool model and the UNCOUPLED switch in the POOL section in the cs-file.
- Unified FLACS-GasEx and FLACS-Fire combustion models.
- DTM radiation can be used in FLACS-GasEx.
- Added DURATION_OF_IGNITION to the IGNITION menu in the cs-file.
- Removed obsolete output variables from the pool model.
- Changes to the mixture model, properties sourced from the species database.
- Improved numerical properties of the leak model at the end of its duration.
- Improved radiation output handling in leak cells.
- CGNS file-locking changed to use the OS file locking mechanism.
- Fixed DPDX units in CGNS file.
- QDOSE added as a 3D output variable.
- Removed obsolete CO output in FLACS-Fire.
- Fixed a bug where wall temperature TWALL was incorrectly interpolated at monitor points
- Fixed a bug where the CFLC and CFLV values read from the dump snapshot were used instead of the values specified in the cs-file.
- Fixed a bug where the TIME value read from the dump snapshot could trigger TSTOP stop criterion.
- Fixed a bug with Heat file handling, where heat fluxes and temperatures were never applied to heat objects.
- Fixed a bug where 3D output variable TDOSE not properly integrated over time, giving wrong results.
- Fixed the velocity profile for WIND boundary condition when set to ZHI.
- Fixed issue with suspend/resume functionality in RunManager for scenarios that load results (dump snapshots) from previous simulations.
• Fixed enthalpy of formation for $SO_2$ that may have led to explosion pressure underestimation for $H_2S$ or other gases and dusts containing Sulphur.

• Fixed negative values of expansion ratio in FLACS-Fire that resulted in occasional crashes and small temperature differences.

• Fixed issue led to slightly different results when running FLACS-Fire simulations in serial and OMP mode.

• Various other minor improvements and bug fixes.

1.6.4.2 CASD

• Added the "Duration of ignition" setting to the scenario menu to specify a period during which ignition should be continuously attempted.

• A new mixtures GUI is available in the Gas composition and volume section of the scenario menu. This gives access to many more species and makes their properties visible.

• Grid wizards are now available to automatically configure grids and CFL numbers for pool and jet fire scenarios.

• Fixed a bug that caused the Leak Wizard to calculate the leak area incorrectly for some cases.

• Various other minor improvements and bug fixes.

1.6.4.3 Flowvis

• Fixed a bug that prevented some variables from being plotted due to the removal of the minimum and maximum data node from the cgns file.

• Fixed a bug that required presentations to be saved to the same network drive as the simulation files.

• Fixed a bug that sometimes made 3D data appear unavailable.

• Improvements have been made to fire and smoke visualisations.

• Plots of radiative heat fluxes now show the data interpolated between the specific solution points, with an option to view the underlying data without the interpolation.

• Removed the cgns lock file so that data can be viewed while simulations are still running.

• Various other minor improvements and bug fixes.

1.6.5 FLACS-CFD 21.2

1.6.5.1 Simulators

• The new FLACSCFD license feature is used for the solver executable encryption and is required to launch any simulation (see FLACS-CFD solver licensing).

• The license control is changed to account for the new per-thread licensing model.

• FLACS-DustEx integrated to the flacscfd solver.

• Added smoke visibility output for fire simulations.

• Advanced heat radiation modelling, that was previously available only for fire simulations, is made available for explosion simulations.
1.6 What is new in this release?

• Fixed a bug in the output variables TDOSE, PDEATH and PROBIT where erroneous (non-conservative) values were calculated due to wrong unit conversion. The bug affected flacscfd but not flacs2.

• Fixed a bug in entraining leaks affecting the mass rate, due to an erroneous location of leak cell and the related sink cell accounting for entrained air. The bug affected flacscfd but not flacs2.

• Fixed a bug where the dump snapshot was not written whenever the dumping time (TDUMP) and stop time (TSTOP), specified in the cc file, coincided.

• Fixed a bug where the dump snapshot was not updated when the dumping time (TDUMP) was repeated in the cc file.

• Fixed an issue where wind buildup time settings affected the convergence detection in the steady-state solver.

• Fixed an issue where interruption and restart of a steady-state simulation affected the convergence detection.

• Fixed a bug where the liquid heat capacity of user-defined species was incorrect (affected only liquid pool simulations).

• Cloud interface: improved accuracy in mapping fuel volume(s) defined in the cloud file to the computational grid.

• Improved convergence properties with decreased occurrence of large mass residuals ("*** MASS RESIDUAL =..." messages).

• Improved informative message on ignition failure in dust explosion simulations (message "IGNITION FAILED!") providing dust concentration at ignition point and flammable range.

• Various minor improvements and bug fixes.

1.6.5.2 CASD

• Scenario section for Radiation is added and available for gas explosion and fire.

• Fixed bug regarding leak types AIR, SUCTION and FAN are not set correctly in the leak wizard.

• Various minor improvements and bug fixes.

1.6.5.3 Flowvis

• Added option for custom oriented slices in 3D plot.

• Transparent background in images exported to GeoTIFF format.

• Various minor improvements and bug fixes.

1.6.6 FLACS-CFD 21.1

This release contains changes to address the following issues:

• FLACS-Cloud support on Amazon AWS.

• Chinese translation of CASD, Flowvis, RunManager, FGC and FLACS-Risk graphical user interfaces.

• Pool model has been improved and moved into the next-generation solver flacscfd.

• The Python API has been improved. See changelog in API reference for details.

• Improved default initial turbulence conditions for gas explosion simulations.
• Flowvis 2D plot shows selected scalar and vector variable values in Verify Porosities tool.
• Performance improvements when reading CGNS files in Flowvis.
• Fixed critical bug: cloud size metrics (e.g. FLAM, Q9 etc) were incorrect (but conservative). Values of these metrics were reported for whole simulation domain rather than for the Gas monitor region as expected.
• Fixed the default reference LFL value (lfl_lev) for the custom gas monitor regions monitoring LFL percentiles.
• Fixed bug in the custom gas monitor regions - user choice of upper (ufl=) and lower (lfl=) flammability limits were not effective.
• Fixed scaling of Line monitor Integral variable when LFL keyword is given.
• Fixed bug where panel variables have no output for Fire output variables.
• Fixed radiative flux calculations at monitor points inside DTM domain.
• Fixed incorrect group assignment of panels in CGNS file and Flowvis.
• Updated license manager software (now using version 8.53).
• Improved automatic convergence checking algorithm to adjust underrelaxation factors towards defaults.
• Various minor improvements and bug fixes.

1.6.7 FLACS-CFD 20.2

This release contains changes to address the following issues:
• Steady-state solver prematurely stopped at 10000 iterations.
• Steady-state solver did not write dump snapshots on convergence.
• Excessive memory usage by simulator for certain cases.
• Long wait time when suspending fire simulations.
• Support package installation of license runtime on Windows.
• Loss of settings for +IMP pressure panels when opening scenario in Casd.
• Leak wizard did not remember leak duration when opening an existing leak or creating a manual leak.
• Simulator crashes when reading a dump file when certain output variables were enabled (e.g. FMOLE_MAX).
• Simulator exited prematurely when loading dispersion dump file into pool model.
• Visualisation differences between Casd and Flowvis.
• Incorrect unit conversion in flash2 utility for partial pressure.
• Incorrect file permissions on installed files when installing as root user on Linux.

1.6.8 FLACS-CFD 20.1

The name of the package has been changed to FLACS-CFD, release of next-generation solver flacscfd and update of some of the main libraries and compilers. See section Compatibility between FLACS v10.9 and FLACS-CFD 20.1 for more information.
1.6 What is new in this release?

1.6.8.1 Simulators

Released next-generation solver flacscfd. The changes to the solver(s) are divided in two sections to make it clear of the difference between flacs2 and flacscfd. For an overview of which variants is available in which version see section flacscfd.

flacscfd:
- New and improved architecture
- Seamless suspend/resume of simulations with self-contained CGNS dump files
- Steady-state solver for faster simulations times for non-transient cases
- Fixed issue with ground roughness
- New and improved pressure relief panel types

flacs2:
- Removed simulator variants now supported by flacscfd

1.6.8.2 CASD

- New and improved species definition using DIPPR
- Ability to give custom names to user species
- Support undo/redo in scenario menu
- New and improved property editing in object window
- Various improvements in the Leak Wizard
- Support for FlashUtility in the Leak Wizard
- Various support for georeferencing
- Improved materials, including support for texturing
- Improved rendering techniques for photorealistic effects
- Material support for terrain
- Export FLACS-CFD geometry to OBJ
- Various improvements to the quick filter
- Changed name of executable from casd.ivf(.exe) to casd(.exe)
- Removed possibility to have Local objects in database

1.6.8.3 Flowvis

- Vector field visualisation in 2D Slice in 3D plot
- Moved options related to 3D volume visualisation
- Set value range based on LFL/UFL
- Define variables in post-processing using filters
- Animation of pressure relief panels
- Functionality for exporting 360 degree videos
- Changed name of executable from flowvis5(.exe) to flowvis(.exe)
- Batch export of video and images
1.6.8.4 FLACS-Risk

- Added autocompletion of keys in the modifier table widget
- Improved explosion cloud placement

1.6.8.5 RunManager

- Show estimated disk usage for each simulation
- GUI elements for suspend/resume of flacscfd simulations
- Right-click to open scenarios in CASD
- FLACS-Cloud related changes
  - User configurable maximum runtime
  - Display cloud usage statistics
  - User selection of FLACS-CFD version to use when running simulations (e.g. v10.9, 20.1)
  - Display CPU/disk quota
  - Ability to download entire projects without any local files
- Removed usage of version number

1.6.8.6 Other

- Geo2flacs support for importing materials
- Implemented real gas laws in the jet utilities
- Various improvements to the FLACS-CFD Python API
- Various improvements to FGC (Flacs Geometry Calculator)

1.7 Compatibility

1.7.1 Compatibility between FLACS-CFD 22.2 and 22.1 and 21.x

The versions are fully compatible, with the following notes:

- New calculation method for mixtures, may change LFL and UFL slightly for mixtures.
- Corrected expansion ratio may change results slightly.
- New Detonation model may increase calculated overpressures for hydrogen explosions or when triggered manually for other gases.
- Changes to FGC may change calculated porosities and results.
- coXXXXXX.dat3 files are no longer required (unified mesh/primitive geometry model coXXXXXX.geo used).
- coXXXXXX.tri file is no longer required (terrain can be included in geometry model).
- The coXXXXXX.dat3 and coXXXXXX.tri files from previous FLACS versions can still be used by the 22.2 solver, however, when using new 22.2 features (such as terrain or instancing) in the geometry file it can no longer be opened in previous versions.
1.7 Compatibility

1.7.2 Compatibility between FLACS-CFD 21.x and 20.2

The versions are fully compatible, with the following notes:

- The CS-file compatibility field have been updated to handle minor changes in the scenario setup files; the changes are backwards compatible.

- The cs*.POOL is no longer needed by the simulator. The information is now written to the cs*.dat3 file by CASD.

1.7.3 Compatibility between FLACS-CFD 20.1 and 20.2

The versions are fully compatible.

1.7.4 Compatibility between FLACS v10.9 and FLACS-CFD 20.1

Some of the major libraries used by the package have been updated to newer versions, more specifically the current releases now uses Qt 5.12 and Python 3. On windows the MSVC 14.1 compiler is used, while on linux GCC 7 is used. This will generally only impact users on older linux distros, e.g. Redhat/CentOS 6 and below.

The biggest change in this release is the new and improved flacscfd simulator. This replaces a number of simulator variants previously included in flacs2. The underlying models in flacscfd are the same as in flacs2, but the architecture is different. In addition to the simulator, a new porosity calculator, FGC (Flacs Geometry Calculator), have been released.

CASD and Flowvis in FLACS-CFD 20.1 can read all scenario and presentation files, respectively, of previous versions. The new features of 20.1 are not supported in previous releases. Files that contain these changes might not be supported by older versions of CASD and Flowvis.

The default value of the VERSION field in the Scenario file have been changed to v20.x. Scenarios that do not have this version will not run in flacscfd.

The installer package have been changed, and will no longer install the license manager runtime software. For details see section License manager runtime software.

1.7.5 Compatibility between FLACS v10.8 and v10.9

The Flacs simulators included in v10.9, include a new gas mixing rule/algorithm for Hydrogen-Inert mixtures. When enabled this rule will calculate more accurate flammability limits and laminar burning velocities of Hydrogen-Inert mixtures. This modification fixes an issue in all previous versions for these mixtures. See Significant overprediction of lower flammability limits for hydrogen+inert mixtures and Hydrogen-Nitrogen specific laminar burning velocity model for more information.

Apart from this change the Flacs simulators included in v10.9 contain only very minor changes to improve robustness and should not change results compared to v10.8.

CASD and Flowvis in v10.9 can read all scenario and presentation files, respectively, of previous versions. The new features of v10.9 (see above) are not supported in previous versions, so that input involving these will not be understood by, for example, CASD, Porcalc and Flowvis of v10.8 or earlier.

The licensing system have been updated in FLACS v10.9. If you are using a network license the licensing software installed on the license server must be updated to the latest version. See section License manager runtime software for information about how this can be done. The newest version of
the licensing software is compatible with all earlier versions of FLACS (i.e. updating the licensing software on the license server should not affect earlier versions of FLACS).

For specific compatibility information related to LNG studies in the US, see section on Compatibility between FLACS v10 and v9.1.

1.7.6 Compatibility between FLACS v10.7 and v10.8

The Flacs simulators included in v10.8r1 contain only very minor changes to improve robustness and should not change results compared to v10.7r2.
CASD and Flowvis in v10.8 can read all scenario and presentation files, respectively, of previous versions.
The new features of v10.8 (see above) are not supported in previous versions, so that input involving these will not be understood by, for example, CASD, Porcalc and Flowvis of v10.7 or earlier.

1.7.7 Compatibility between FLACS v10.6 and v10.7

CASD and Flowvis in v10.7 can read all scenario and presentation files, respectively, of previous versions.
The new features of v10.7 (see above) are not supported in previous versions, so that input involving these will not be understood by, for example, CASD, Porcalc and Flowvis of v10.6 or earlier.

FLACS-Risk
The algorithm for setting simulation job numbers has been made more robust. Because of the changes, when a pre-v1.1r3 Risk project is opened and re-exported with FLACS-Risk v1.1r3, the simulation job numbers may change and may no longer be consistent with previously calculated result files. FLACS-Risk with indicate this, but it may require rerunning most simulations. If you have old projects with many results, please use one of the following workarounds:

1. Open the project in FLACS-Risk v1.1.r2.
2. Open the project read-only. Note that this is only an issue when a re-export is done. If the existing project is only opened, it will keep the existing job numbers.

1.7.8 Compatibility between FLACS v10.5 and v10.6

CASD and Flowvis in v10.6 can read all scenario and presentation files, respectively, of previous versions.
The new features of v10.6 (see above) are not supported in previous versions, so that input involving these will not be understood by, for example, CASD, Porcalc and Flowvis of v10.5 or earlier.
The Flacs simulators are included in v10.6 without changes compared to v10.5, except the Flacs-Fire solver: Due to the introduction of the (default) automatic DTM domain and far-field models, simulations rerun in FLACS v10.6 will likely be somewhat different, especially in the far field. In most cases the differences are expected to be modest. It is still possible to enable the full DTM domain (using the DTM domain constraint setting) in v10.6, which should give very similar results to those obtained in v10.5. For the variable QWALL, significantly lower values may be obtained; previous results were excessively conservative.

1.7.9 Compatibility between FLACS v10.4 and v10.5

CASD and Flowvis in v10.5 can read all scenario and presentation files, respectively, of previous versions.
The new features of v10.5 (see above) are not supported in previous versions, so that input involving these will not be understood by, for example, CASD, Porcalc and Flowvis of v10.4.
The Flacs simulators are included in v10.5 without changes compared to v10.4r2.

1.7.10 Compatibility between FLACS v10.4r1 and v10.4r2

Concerning file formats etc. v10.4r2 and the predecessor v10.4r1 are fully compatible. Flacs results should only change when monitor point output from Flacs-Fire is considered; here, the differences can be significant due to the improved model implemented in v10.4r2.
1.7 Compatibility

1.7.11 Compatibility between FLACS v10.3 and v10.4

The simulators included in the FLACS v10.4 package have undergone minor improvements and bug fixes. Changed simulation results have to be expected mainly due to several corrections in Porcalc, which is included with a new default version 2.8. This version is recommended, but for backwards compatibility in ongoing projects, the previous versions are included in the package. The magnitude of differences that can occur in simulation results due to the improvements in Porcalc is exemplified below.

The Flowvis versions 4.4 and 5.3, which are both included in FLACS v10.4, use different file formats for presentations. A presentation created with Flowvis version 4.4 cannot be handled in version 5.3 and vice versa.

1.7.11.1 Changes in results due to bug-fixes in Porcalc

As mentioned above, several recently discovered bugs have been fixed in Porcalc for FLACS v10.4r1. While the modelling principles are the same as before, the elimination of those mistakes in the implementation does lead to changes in the results of Porcalc and consequently also Flacs simulations.

To quantify the changes in the results, Gexcon has re-run the test suite of approximately 950 simulations that has been used for this purpose before. The simulations concern 15 different geometries, ranging from medium-scale experiments to full scale on- and offshore installations. The key output parameters of this test are the maximum overpressures in a number of monitor points for each scenario. The differences in the results arise indirectly due to changed turbulence generation and flame folding parameters based on the geometry and are shown in the plots below.

On closer examination, some of the bigger deviations can be attributed to a geometry that has a number of cylinders that are aligned with the grid in exactly the way that was affected by the implementation bug, and the ignition location is situated close to these obstructions. In this constellation the effect in turbulence generation and flame acceleration is particularly emphasised.

The effect of nested primitives should be minor for custom-built geometries and those filtered properly in the CAD export. The change is relevant (and has been discovered) when importing complex/as-built geometries into FLACS. For such geometries with many nested primitives the effect can be considerable, with the uncorrected versions of Porcalc tending to overestimate the explosion effects.

Objects inside empty space created by a left difference operation created incorrect porosity patterns in the Porcalc version prior to 2.8. This should be a relatively rare situation and not play any significant role for most practical cases.

In summary, the software changes between Porcalc 2.7.1 and 2.8 must be expected to entail noticeable changes in the results of FLACS simulations. These changes are due to the corrections of mistakes in the modelling and should therefore be seen as improvements.

1.7.12 Compatibility between FLACS v10.2 and v10.3

- The simulators included in the FLACS v10.3r2 package are fully compatible with the previous release, v10.2r2, and results should not change when simulated with these two versions on the same hardware and using identical input files, except for the following types of scenarios:
  - when a setup file with the name cs<JOBNO>.SETUP is present that changes the default behaviour of the simulator but was not used when running with v10.2r2,
  - wrongly specified (too big) area leaks will trigger errors rather than warnings.

  Except for scenarios that are affected by the above changes, no differences in results are to be expected. Gexcon's test suite has confirmed that only very minor numerical variations occur. These may be caused by compiler or optimisation changes, operating system or hardware differences etc.

- The Flowvis versions 4.4 and 5.1, which are both included in FLACS v10.3r2, use different file formats for presentations. A presentation created with Flowvis version 4.4 cannot be handled in version 5.1 and vice versa.
Introduction

• The interpolation algorithm for visualisation of surface values in 3D plots has been significantly improved in Flowvis 5.1 included in FLACS v10.3r2. It now accounts for porosities and alignment of large non grid aligned objects. Previously, the interpolation algorithm used only linear interpolation without accounting for porosities and solid walls. So in many cases this resulted in lower values (by up to a factor of 2\times) being shown. The new algorithm is enabled by default. For legacy reasons, the previous interpolation algorithm from FLACS v10.2r2 is also still available by selecting 'Fast interpolation' in the menu. However, it is strongly recommended to no longer use the previous FLACS v10.2r2 version or the legacy 'Fast interpolation' option.

Note: This change only applies to 3D surface plots. Therefore 3D volume plots and 2D cut plane plots are not affected and should not show differences between FLACS v10.2r2 and FLACS v10.3r2.

1.7.13 Compatibility between FLACS v10.1 and v10.2

• The simulators included in the FLACS v10.2r2 package are fully compatible with the previous releases since v10.1r1, and results should not change when simulated with these versions on the same hardware and using identical input files, except for the following types of scenarios:
  – Flacs-Explo will give slightly different (more correct) results due to a bug-fix which improves symmetry for symmetric scenarios; the fix was first delivered with v10.2r1 and also applies to other Flacs simulator variants when using blast blocks.

• The Flowvis versions 4.4 and 5.0, which are both included in FLACS v10.2, use different file formats for presentations. A presentation created with Flowvis version 4.4 cannot be handled in version 5.0 and vice versa.

1.7.14 Compatibility between FLACS v10.0 and v10.1

The FLACS v10.1r1 package is fully compatible with the previous release, v10.0r1. Results should not change when simulated with these two versions on the same hardware and using identical input files, with the following two exceptions:

• Dispersion scenarios with a non-constant vertical wind and temperature profile may yield different results close to the outflow boundaries; v10.1 has been re-aligned with v9.1r4 regarding the vertical profiles at the inflow. If the v10.0 parameterisations are desired then the compatibility key "ABL=f240" can be used.

• In simulations using the incompressible solver together with quiescent or low-wind initial conditions, the length of the first time step is more rigorously limited than in previous versions of FLACS.

1.7.15 Compatibility between FLACS v10 and v9.1

Scenarios that have been created in FLACS v9.1 can be run in FLACS v10. However, the porosities for the v9.1 scenarios should be recalculated with the Porcalc included in the latest FLACS v10 package before running them with the v10 simulator. This is due to an improvement in the porosity calculation by Porcalc, with a related change in the production of sub-grid turbulence. You can expect to see slight changes in the results compared to FLACS v9.1. In validation runs for hydrogen, and other highly reactive fuels, more accurate and reliable over-pressures were obtained. In simulations of the BFETS and HSE modules (with different congestion levels and ignition points) the results were nearly unchanged.

It is possible to use the sub-grid turbulence production model of FLACS v9.1r4 in FLACS v10 by using the key STF=f228. When this key is used, the log file (rt.dat3, or the tt-file on a Linux computer) will contain the line # SUBGRID TURBULENCE FACTOR STF="f228:0.600", which indicates that the sub-grid turbulence production model of FLACS v9.1r4 (simulator version flacs2.2.8) is used, even though the log file contains the current simulator version, e.g. FLACS, Version 2.5.2, July 2014, Gexcon AS.
1.9 Feedback from users

Feedback on the content in this manual is most welcome, and FLACS-CFD users may submit their comments or suggestions by e-mail to: flacs@gexcon.com
When submitting comments or suggestion to the content of the manual, or when pointing out misprints in the text, please indicate the relevant page numbers or sections, and the corresponding version of the manual (date issued).
Chapter 2

Getting started

This chapter describes the basics of setting up the FLACS-CFD software for new users. It includes recommendations concerning the user threshold, typical hardware requirements, and procedures for installing FLACS-CFD on both Linux and Windows.

2.1 Prerequisites for users

Efficient use of FLACS-CFD does not require detailed knowledge about computational fluid dynamics (CFD). However, you should have some experience in the application of computers for routine tasks, such as text editing. The proper interpretation of simulation results requires adequate knowledge within the field of fluid dynamics. A suitable starting point for the novice in the field of gas explosions is the Gas Explosion Handbook (Bjerketvedt et al., 1993) from Christian Michelsen Research (CMR), and new users of FLACS-CFD should attend a four-day introductory course arranged by Gexcon AS.

2.2 Hardware and software requirements

FLACS-CFD is available on Linux and on Microsoft Windows. The hardware requirements for running the FLACS-CFD software depend to some extent on the size of the problem in question, that is, the number of grid cells required to resolve the computational domain properly. Most modern computers, both desktops and laptops, will perform well for small or medium sized problems. For simulating large problems with FLACS-CFD, “the more the better” applies to both processing speed and memory (RAM) size. Storing of large amounts of simulation data increases the requirements for disk space.

See also:

For detailed and updated hardware and software requirements, including a list of operating system versions that FLACS-CFD 22.2 has been tested on, please check the FLACS-CFD technical requirements page.

A powerful graphics card with OpenGL driver supporting OpenGL version 3.3 or higher is recommended for CASD and Flowvis. Older OpenGL versions may work (2.1 is the minimum), but if insufficient OpenGL support is detected, not all functionality will be available, e.g. no 3D views/plots. It is recommended to use the latest graphics driver version available for the target hardware. Check the web site of your hardware vendor for relevant updates.

Using the “nouveau” open-source NVIDIA driver can cause problems with the 3D plot. It is recommended to use the official drivers from NVIDIA.

Some compatibility issues on older or built-in graphics cards can be related to lacking or improper support for OpenGL instancing. If you experience crashes when creating a 3D plot with geometry then try setting the environment variable OSG_DISABLE_INSTANCING, that is,
• on Linux: open a shell and, if you use the csh/tcsh shell, type

```bash
> setenv OSG_DISABLE_INSTANCING 1
```

or, if you use the bash shell, type

```bash
> export OSG_DISABLE_INSTANCING=1
```

• on Windows: open a command prompt and type

```bash
> set OSG_DISABLE_INSTANCING=1
```

or set the variable via the Environment Variables dialogue.

**Note:**

Systems that have multiple graphic processing units (GPU), e.g. a NVIDIA GPU in addition to
a Intel chipset GPU, may experience an issue in Microsoft Windows with CASD and Flowvis
not utilizing the powerful NVIDIA GPU. This can be due to performance and/or power settings
in Windows, and can be solved by setting graphics preference to High Performance in the
Advanced Graphics Setting in Windows Display settings. Another option can be to disable
dynamic GPU switching, or to right click on the CASD/Flowvis executable, select Properties
and look for options to select which GPU to use.

Running FLACS-CFD on computers with high resolution screens may result in challenges with the
user interface related to font size and icon size. Setting the environment variable
`QT_AUTO_SCREEN_SCALE_FACTOR=1` might improve the size of user interface items like icons.
On Windows 10 it is possible to right-click on executables in the install folder (e.g. casd.exe) and
select "Properties" at the bottom of the context menu (see image below). In the "Properties" dialog,
switch to the second tab name "Compatibility", and click on "Change high DPI settings". In the
dialog that opens, check the "Override high DPI scaling behaviour...", and select "System" from the
drop down.

### 2.3 Software installation and setup

A valid license is necessary for running FLACS-CFD. This section describes how to install the software,
including the license manager runtime software, and how to activate the license.

FLACS-CFD is distributed in a single setup file for 64 bit operating systems. Make sure to select the correct
installation package for your operating system:

- **Linux**: `flacs-cfd-22.2-linux-64bit-installer.bin`
- **Windows**: `flacs-cfd-22.2-windows-64bit-installer.exe`

**Attention:**

The directory and file names inside the installation directory `Gexcon` and sub structure must not be
changed as this will prohibit FLACS-CFD from working as intended.

**Note:**

The license manager runtime software is only required on machines that have a hardware key connected
or softlock key, see section License manager runtime software for more information.

FLACS-CFD-22.2 requires license manager software version 8.53 or higher. It is recommended to
update your license server(s) with the latest version (currently 8.53). An error message "Sentinel
License Manager version too old" or "HASP LM version is too old" would indicate that the license
manager software on the license server(s) needs to be updated.
2.3 Software installation and setup

2.3.1 On Linux

When FLACS-CFD is installed on a Linux system it can either be made available to a single user or to all users.

2.3.1.1 Installing for a single user

If only one person will be using FLACS-CFD, the software can be installed in this user's home directory. FLACS-CFD will by default be installed under /home/my_user/Gexcon. Follow the instructions given. It is recommended to keep the default parameters.

1. Save the installation package to a convenient location.
2. Make sure the file is executable: Type
   
   ```
   > chmod u+x /home/my_user/flacs-cfd-22.2-linux-64bit-installer.bin
   ```

3. Run the installation program: Type

   ```
   > /home/my_user/flacs-cfd-22.2-64bit-linux-installer.bin
   ```

   Instructions on setting up the FLACS-CFD license manager are below in the section Setting up the FLACS-CFD license.

2.3.1.2 Installing system wide as super user

Access to the system super user (“root”) is required to install FLACS-CFD system wide. The FLACS-CFD installation package should be located in the directory, /path/to/installation.

Follow the instructions given. It is recommended to keep the default parameters.

1. Change user to super user (“root”): Type

   ```
   > su
   ```

   and provide the root password as required.

2. Make sure the file is executable: Type

   ```
   > chmod u+x /path/to/installation/flacs-cfd-22.2-linux-64bit-installer.bin
   ```

3. Run the installation program:

   ```
   > /path/to/installation/flacs-cfd-22.2-linux-64bit-installer.bin
   ```

   Instructions on setting up the FLACS-CFD license manager are below in the section Setting up the FLACS-CFD license.

2.3.1.3 Unattended installation

Unattended installation is not supported. The simplest solution is to install as normally on one machine, and then create an archive of the installation directory, which can be extracted on any other machine where the software should be installed. You can also contact Gexcon support to request a archived version of the installer.

It is also possible to install FLACS-CFD on a network filesystem, which will allow the software to be used from any computer having access to the filesystem.
2.3.2 On Windows

To install FLACS-CFD on Windows double-click the installation package
flacs-cfd-22.2-windows-64bit-installer.exe. This will start the installation wizard. Follow
the instructions given. It is recommended to keep the default parameters.
Instructions on setting up the FLACS-CFD license manager are below in the section
Setting up the FLACS-CFD license.

2.3.2.1 Unattended installation

Unattended installation is not supported.
The simplest solution is to install as normally on one machine, and then create an archive of the installation
directory, which can be extracted on any other machine where the software should be installed. You can also
contact Gexcon support to request a archived version of the installer.
It is also possible to install FLACS-CFD on a network filesystem, which will allow the software to be used
from any computer having access to the filesystem.

2.3.3 Setting up the FLACS-CFD license

The license manager runtime software (runtime) need to be installed if you have a hardware key connected
(HL), use a softlock key (SL) or cloud license (CL). The runtime can be automatically installed by the
package installer. See instructions below for manually installing the runtime.

2.3.3.1 License manager runtime software

The license manager runtime software (runtime), when installed, runs in the background. It has a web-based
administration system that can be reached by opening http://localhost:1947 in a web browser.
Installers for the runtime can be found in the installation directory under the license/Sentinel subdirectory.
To install the runtime on Linux use the following steps:

1. See the readme file for detailed instructions (extract readme.tgz and open Readme.html in a web
   browser)

   > mkdir tmp-extract
   > cd tmp-extract
   > tar -xvf
   /usr/local/Gexcon/FLACS-CFD_22.2/license/Sentinel/linux/readme.tgz

2. Extract the package

   > tar -xvf
   /usr/local/Gexcon/FLACS-CFD_22.2/license/Sentinel/linux/aksusbd-8.53.1.tar.gz

3. Run the installer script as root (e.g. using sudo)

   > (cd aksusbd-8.53.1 && chmod -R a+rX . && sudo ./dinst)

Note that in order to support softlock and cloud license keys the Gexcon vendor libraries "haspvlib∗102648.so"
have been included in the same folder as the installer script "dinst” and they must be present there during
installation.
To install the runtime On Windows use the following steps:

1. See the readme file for detailed instructions (extract readme.zip and open Readme.html in a web
   browser)

   > mkdir tmp-extract
   > cd tmp-extract
   > C:\Program Files\7-Zip\7z.exe x'
   C:\Program Files\Gexcon\FLACS-CFD_22.2\license\Sentinel\windows\readme.zip
2.3 Software installation and setup

2. Run the installer as administrator

> C:\Program Files\Gexcon\FLACS-CFD_22.2\license\Sentinel\windows/haspdinst.exe -i

Options:

- install (or -i) option to install
- remove (or -r) option to remove
- info option to show information about contents
- help option to show a brief help

Note that in order to support softlock license keys the Gexcon vendor library "haspvlib_102648.dll" has been included in the same folder as the installer "haspdinst.exe" and it must be present there during installation. The license manager programs running on the computer are called aksusbd and hasplmd on Linux, and hasplms.exe on Windows.

2.3.3.2 Per user configuration of license

Configuration of the license per user can be done via the hasp_104628.ini file. This is useful if you want to access specific license servers.

Note:

The hasp_104628.ini file only affects licenses from Gexcon, and it is optional (normally not required to make the license work).

The location of the hasp_104628.ini file is:
On Linux:

`~/.hasplm/hasp_104628.ini`

On Windows:

`C:\Users\USER_NAME\Appdata\Local\SafeNet_Sentinel\Sentinel LDK\hasp_104628.ini`

Example hasp_104628.ini file to use only specific license servers

```
broadcastsearch = 0
serveraddr = 10.0.0.1
serveraddr = 10.0.0.2
disable_IPV6 = 0
```

Example hasp_104628.ini file to use only specific cloud license (CL) client identity

```
broadcastsearch = 0
serveraddr = UNIQUE:SecretClientIdentity@cloud-license.gexcon.com
disable_IPV6 = 0
```

Example hasp_104628.ini file to use any detected license server

```
broadcastsearch = 1
aggressive = 1
disable_IPV6 = 0
```

2.3.3.3 Single user license

Simply connect the license key to the USB slot on the computer and FLACS-CFD is ready for use.
2.3.3.4 Network license

Follow the same steps as for Single user license on the computer that will hold the FLACS-CFD license. Other computers on the network should automatically detect this license and FLACS-CFD is ready for use. If it does not work, the licenses may not be available from the client computer. On a client computer with the license manager runtime installed, to see the licenses available on the client, open \texttt{http://localhost:1947} in a web browser. If it does not show any FLACS-CFD license keys, then the IP address of the computer having the license key needs to be added. Go to the configuration page, and open the tab Access to Remote License Managers. Make sure Allow Access to Remote Licenses is checked, and enter the IP-address of the computer having the license key in the Remote License Search Parameters box.

See also:

The section Common problems has extra information if the above suggestions do not work, and for Windows client computers, which should be able to run without the license runtime.

2.3.3.5 Cloud license (CL)

The CL license server is hosted by Gexcon and all licenses installed there are maintained by Gexcon. The user is identified by the so-called client identity - a secret string giving access to the purchased license features. The client identity must be added on all client computers to the list of Remote License Search Parameters in the Admin Control Center web interface as it is done for network licenses (see Network license), or in the configuration files as explained in Per user configuration of license.

2.3.3.5.1 Availability

The CL license server is hosted by Gexcon on servers of Amazon AWS (Amazon Web Services). The implementation follows the guidelines for a highly available cloud license server as specified in this document by the licensing solution vendor. It includes, among others, the following measures:

- 2 redundant license server instances in different physical locations;
- 2 redundant database server instances in different physical locations.

2.3.3.5.2 Technical requirements

Client machines must have internet access and outgoing connections over HTTP (port 80), HTTPS (port 443) and over port 1947 must be possible to address: \texttt{cloud-license.gexcon.com}.

2.3.3.5.3 Troubleshooting the connection to the cloud license server

Before contacting support it is strongly advised that the user tests the network connection to the license server. Test whether ports 80, 443 and 1947 are accessible using the OS telnet command:

\texttt{> telnet cloud-license.gexcon.com <PORT>}

Positive result: No output, terminal goes black. Hit enter to disconnect.

Negative result:

Connecting To cloud-license.gexcon.com...Could not open connection ...

On negative result try disconnecting from the company VPN. Test connection again, if works fine then contact your IT support to enable access to \texttt{cloud-license.gexcon.com} via respective port over VPN.

2.3.3.6 Updating/renewing a license

When a license update or renewal is requested, you will receive it in the form of a V2C file from Gexcon. You can apply the update in one of three ways:
2.3 Software installation and setup

2.3.3.6.1 Using the Admin Control Center web interface

1. Open \url{http://localhost:1947} (or \url{http://<IP-address>:1947} if the license key is on a different computer).
2. Go to the Update/Attach page.
3. Select the V2C file received from Gexcon, and apply it.

2.3.3.6.2 Using the FLACS-CFD Configuration Wizard

This needs to be done on the computer the license key is installed on.

1. Start the Configuration Wizard (change paths if FLACS-CFD is installed in a different location):
   - Linux: Type
     \$ /usr/local/Gexcon/FLACS-CFD_22.2/bin/run configureWizard
   - Windows: Type
     > C:\Program Files\Gexcon\FLACS-CFD_22.2\bin\configureWizard.exe
2. Click Apply license update.
3. Load the update file.

2.3.3.6.3 Using the FLACS-CFD command line license configuration tool

The following needs to be done on the computer the license key is installed on.
Run the command line tool with the update option (change paths if FLACS-CFD is installed in a different location):

   - Linux: Type
     > /usr/local/Gexcon/FLACS-CFD_22.2/bin/licenseconfig -u <path to V2C file>
   - Windows: Type
     > C:\Program Files\Gexcon\FLACS-CFD_22.2\bin\licenseconfig.exe -u <path to V2C file>

2.3.3.7 Cancelling a license

When a license cancellation is requested, you will receive it in the form of a V2C file from Gexcon, and must apply this one in the same way as a license update, described under Updating/renewing a license. In addition, a confirmation that the license cancellation has been carried out needs to be sent to Gexcon. This confirmation is in the form of a C2V file, containing license information. This file can be created in two ways:

2.3.3.7.1 Using the FLACS-CFD Configuration Wizard

The following needs to be done on the computer the license key is installed on.

1. Start the Configuration Wizard (change paths if FLACS-CFD is installed in a different location):
   - Linux:
     \$ /usr/local/Gexcon/FLACS-CFD_22.2/bin/run configureWizard
   - Windows:
2. Click Retrieve license information.

3. Save the license information to a file.

4. Mail this file to flacs@gexcon.com.

2.3.3.7.2 Using the FLACS-CFD command line license configuration tool The following needs to be done on the computer that has the license key installed.

1. Run the command line tool with the info option (change paths if FLACS-CFD is installed in a different location):

   Linux:

   `/usr/local/Gexcon/FLACS-CFD_22.2/bin/licenseconfig -o -i`

   This writes the license information to a file with default name default.c2v. The user can optionally specify the name of the output file. And if there are more than one license key on the computer, the user need to specify one of the keys. If the user e.g. wants license information for key with ID 208312663, and the name 208312663.c2v for the output file:

   `/usr/local/Gexcon/FLACS-CFD_22.2/bin/licenseconfig -o 208312663.c2v -i`

   In a similar way for Windows:

   `C:\Program Files\Gexcon\FLACS-CFD_22.2\bin\licenseconfig.exe -o -i`

   This writes the license information to a file with default name default.c2v.

   2. Mail this file to flacs@gexcon.com.

To have an overview of all the options for the tool (Linux, similar for Windows):

```
/usr/local/Gexcon/FLACS-CFD_22.2/bin/licenseconfig -h
```

Gexcon License Configuration Program

`Copyright 2019, Gexcon AS`

Allowed options:

- `-h` [ `--help` ] Show options
- `-i` [ `--info` ] Create license information file (*.c2v)
- `-o` [ `--output` ] Write output to specified file
- `-f` [ `--fingerprint` ] Get machine fingerprint (*.c2v)
- `-d` [ `--diagnostics` ] Get license diagnostics

FLACS-CFD v22.2 User's Manual
2.3 Software installation and setup

2.3.3.8 Checking the number of current sessions

The Admin Control Center can be used to check the limits, and the current number of logins and sessions of the network/local key. Open http://localhost:1947 (or http://<IP-address>:1947 if the license key is on a different computer) and go to the Sentinel Keys page. Select the (Net) Features button. This function lists the limit of concurrent users for CASD, Flacs and Flowvis and also the number of logins for the chosen key. View the Sessions page to see the current sessions running for the chosen network/local key.

If, for some reason, a CASD, Flacs or Flowvis instance has not been closed properly and still shows as a session, then the Disconnect button can be used to force the program to end.

2.3.3.9 Common problems when setting up the license

• The license key is visible in the web interface but FLACS-CFD cannot find it. (Windows)
  Due to a problem in the SafeNet system, the IP-address specified in the web interface is not always used. It must then be specified in a separate .ini file.
  
  1. On Windows XP, create a file called
     
     c:\Documents and Settings\USER_NAME\Local Settings\Application Data\SafeNet Sentinel\Sentinel LDK\hasp_104628.ini
     
     or, on Windows 7/8/10, where USER_NAME is your user name:
     
     C:\Users\USER_NAME\AppData\Local\SafeNet Sentinel\Sentinel LDK\hasp_104628.ini
     
     2. The file must contain the lines:
     
        [NETWORK]
        broadcastsearch = 0
        serveraddr = 10.0.0.1
        disable_IPv6 = 0
     
     where 10.0.0.1 is the IP-address of the license server.

• No license is found even after the license server IP-address has been specified.
  Make sure there is no firewall or antivirus blocking the communication. Port 1947 on the license server must be reachable over TCP and UDP.

• A license server must be specified but the license manager web interface is not reachable.
  (Linux)
  Sometimes Flacs simulations are run on a Linux machine with no display, and the license cannot be set up in the web interface. In this case, you can specify the license server IP in the configuration file /etc/hasplm/hasplm.ini.

  1. Create the file if it does not exist.
  2. The file must contain the lines:

     [REMOTE]
     broadcastsearch = 1
     serversearchinterval = 30
     serveraddr = 10.0.0.1

    where 10.0.0.1 is the IP-address of the license server. You do not need to restart the license manager after editing this file.
2.3.4 Setting up the FLACS-CFD environment

After installation, FLACS-CFD programs can be accessed from the system menu, in the following locations:

**Linux (KDE):** Start → Applications → Edutainment → Construction

**Linux (Gnome):** Applications → Other

**Windows:** Start → All Programs → Gexcon → FLACS-CFD 22.2

Some systems may require the user to log out and restart before FLACS-CFD will appear in the system menu.
Desktops that do not follow the freedesktop.org standards will not install an icon in the Applications menu. This will happen on older distributions. In these cases, you may be able to install icons and associations manually. Contact your GNU/Linux distribution vendor for details on how to customise your desktop.

2.3.4.1 FLACS-CFD User setup on Linux

For easy access to FLACS-CFD from the command line add the following text to your startup file.
If you use the csh/tcsh shell, edit or create the .cshrc file adding

```
alias run /usr/local/Gexcon/FLACS-CFD_22.2/bin/run
```

If you use the bash shell, edit or create the .bashrc file adding

```
alias run=/usr/local/Gexcon/FLACS-CFD_22.2/bin/run
```

FLACS-CFD programs can be started by typing the name of the program, for example, `run flowvis`.

2.3.5 Uninstalling FLACS-CFD

**Linux:** Run the program `/usr/local/Gexcon/uninstall-Gexcon.sh`.

**Windows:** FLACS-CFD can be uninstalled using Control Panel/Add or Remove Programs.

2.4 A Tour of FLACS-CFD

To try the following example, copy the necessary files from the FLACS-CFD installation directory to a work directory (where you have write access):

**On Linux:**

1. Create a new directory (DIRECTORY_NAME) in which you perform the exercise: Type

   ```
   > mkdir DIRECTORY_NAME
   ```

2. Move into this directory: Type

   ```
   > cd DIRECTORY_NAME
   ```

3. Copy the files from the FLACS-CFD installation directory (notice the space before the ".") Type

   ```
   ```

**On Windows:**

1. Start the Windows Explorer, find the drive and directory where you want to create the work directory, right-click, choose New → Folder, and then type the name of the new directory.

2. Copy all files from
A typical simulation session with FLACS-CFD involves several steps. You can initiate a FLACS-CFD session by clicking the FLACS-CFD icon on the desktop:

Figure 2.1: The FLACS-CFD icon

This will open the RunManager window:

Figure 2.2: The FLACS-CFD RunManager.

Some of the main tasks of the RunManager are:

- Starting the preprocessor CASD
- Running CFD simulations
- Starting the postprocessor Flowvis

In the RunManager, click the CASD icon to open the pre-processor.

Figure 2.3: The CASD icon
1. In CASD, open the file 000000.caj
2. Click View, click XZ View and then use the mouse wheel to zoom in on the image.
3. Click Geometry, then click Database, then click Connect, then click GexconPlatform.db, then click CASD_DB and click Open.
4. In the Database, click to expand the entry Nice, then click to expand the entry one,
5. In the Visible column, double-click the check-box next to the entry one.

The Database window looks like this:

![Database window](image)

Figure 2.4: FLACS-CFD preprocessor CASD

Typical tasks performed from the Database window include:

- Creating a new database and new geometries
- Opening existing databases and geometries
- Creating new materials (for example, colours), or modifying existing materials
- Creating new objects, or modifying existing objects

You can build complex objects by adding or subtracting several simple components (for example boxes and cylinders). Any geometry can consist of one or several objects, or assemblies of several objects. An alternative way of working with geometries involves geometry import using the geo2flacs utility. This requires a representation of the geometry in a compatible CAD format (typically Microstation or PDMS). Apart from geometry building, the menus in CASD also allow the following tasks to be performed:

- Definition of the computational domain and the computational grid
- Porosity calculations using the utility program FGC (Flacs Geometry Calculator), as well as porosity verification
- Scenario setup, including:
  - Definition of monitor point locations, and selection of output variables
  - Specification of boundary conditions
– Specification of vent panels and leaks

– Specification of fuel type

– Specification of ignition position as well as time and duration of ignition

After defining the scenario, the next step is to run the actual FLACS-CFD simulations:

• Simulations can be started and monitored with the RunManager

• The same operations can be controlled from the command line in Linux

> run flacs 000000

The RunManager also monitors the simulations while they are running.

RunManager should still be open.

1. Click Add Directory and navigate to the directory with the copied scenario files, and then click Select Folder.

2. Click to expand the directory.

3. Click on the checkbox in the "Job number" column next to the first job in the list (job number 000000), and click simulate.
FGC will first take several minutes to calculate the porosities, then the simulation will begin. The simulation runs for approximately 1.0 s simulated time, and you can follow the progress of the simulation by viewing the time scale at the bottom of the RunManager window.

The final step in a FLACS-CFD session is typically the presentation and verification of simulation results with the postprocessor Flowvis, as well as data extraction and reporting. You can start the postprocessor when clicking the Flowvis icon in the RunManager:

Figure 2.6: The Flowvis icon.
1. Click File, click Open, and navigate to the directory with the copied example files, choose the file presentation000000.fvp and click open.

2. Use the tabs on the right to see the different examples of output that are available.

Figure 2.7: FLACS-CFD postprocessor Flowvis

Some of the most frequently used features in Flowvis include:

- Verifying porosities in a geometry
- Creating scalar-time plots, 2D-plots, 3D-plots, ...
- Creating animations

Data reporting may also include the extraction of numerical simulation results with the utility programs r1file and r3file. These programs can be run from the command line or through the graphical user interface gutils.

2.5 Help and support

FLACS-CFD users can get technical support by contacting the Gexcon software department:

Email: flacs@gexcon.com

Phone: +47 55574330

Professional discussions regarding the use of FLACS-CFD also take place in the FLACS User's Group on LinkedIn.

2.6 Introductory example

This chapter contains an introductory example. It gives a first impression of how to set up and run a simple FLACS-CFD explosion simulation. For additional examples see, e.g., the FLACS-CFD Best practice chapter.
2.6.1 Things to keep in mind

FLACS-CFD is a Computational Fluid Dynamics based Explosion Simulator tool. The input to a CFD calculation is:

- A geometry, either created manually for the specific purpose, or imported from a CAD system
- A grid which divides the simulation domain into cells. Variables such as pressure do not vary in space within a grid cell. FLACS-CFD uses a regular, Cartesian grid, which means box grid cells.
- Various scenario parameters, such as boundary conditions, monitor point locations, gas cloud size, position and composition, and ignition location.

All of the above are normally handled in the FLACS-CFD pre-processor CASD. The geometry is saved to a database. The database file structure starts in a top level directory given a name with the suffix ".db". The database directory should not contain user files, or files other than those created by the database interface in CASD.

In addition to the database, a number of other files are created before and during the simulation. All file names contain the job number, a 6 digit number. The following files are created as input to the simulation (010101 is the job number).

- `cg010101.dat3` The grid file.
- `cs010101.dat3` The scenario file.
- `co010101.geo` The obstruction file. This file contains the geometry model.
- `cp010101.dat3` The porosity file, which is created by FGC (Flacs Geometry Calculator).

During the simulation a set of result files will be created:

- `1010101.cgns` CGNS file - scalar and field output from flacscfd solver
- `r1010101.dat3` Scalar-time output file (Flacs2 only) - output from monitor points.
- `r3010101.dat3` Field output file (Flacs2 only). Needed to create 2D and 3D plots.
- `rt010101.dat3` Simulation log file - rt-file

FLACS-CFD can also create and use other files. See section Files in FLACS-CFD for details.

Each simulation creates several result files and therefore it is important to create a good directory structure to keep track of the files. Details on this and further recommendations are given in section Files in FLACS-CFD.

2.6.2 Creating the work directory

As FLACS-CFD creates a relatively large number of files it is important to have a good system for bookkeeping. It is recommended to start out with an empty directory.

2.6.2.1 On Linux

1. Create a new directory (DIRECTORY_NAME) in which you perform the exercise: Type
   
   > mkdir DIRECTORY_NAME

2. Move into this directory: Type
   
   > cd DIRECTORY_NAME

3. Copy the geometry files (notice the space before the "."). Type
   
   > cp /usr/local/Gexcon/FLACS-CFD_22.2/doc/examples/ex01_explosion/ +00001*. 

4. Start up the FLACS-CFD RunManager: Type
   
   > run runmanager
2.6 Introductory example

2.6.2.2 On Windows

1. Start the Windows Explorer, find the drive and directory where you want to create the work directory, right-click, choose New → Folder, and then type the name of the new directory.

2. Copy the files from
   C: \ Program Files \ Gexcon \ FLACS-CFD_22.2 \ doc \ examples \ ex01_exlosion \ ∗00001* (00001 means all files containing the text "00001").

3. Click the Flacs RunManager icon.

2.6.3 Starting the preprocessor CASD

1. On the Tools menu, click Casd.

2.6.3.1 Opening and viewing the geometry in CASD

1. In Casd, click File, click Open, and navigate to the directory you created in the section above.

2. Choose 100001.caj, and click Open.

3. If any error message appears click OK.

4. Use the mouse to examine the structure.

![Figure 2.8: The geometry used in the example.](image)

2.6.3.2 Making a grid for the simulation

The structure shown has dimensions of 25.6m x 8m x 8m; the origin is in the left front lower corner.

1. Click Grid, then click Quick Grid.
2. Enter the numbers as shown in the screenshot.

![Quick Grid interface](image)

Figure 2.9: Creating the grid in Quick Grid

3. Click Grid, then click Information. Confirm that the control volumes are all $0.53 \times 0.53 \times 0.53$ m, then click OK.
2.6 Introductory example

4. Click Porosities, then click Calculate.

5. Click Grid, then click Display Off.

2.6.3.3 Defining an explosion scenario

1. In Scenario Settings, in the Simulation type drop-down list, click Gas explosion.

2. In the scenario menu click the arrow to the left of Monitor Points to expand it.

3. Click Add, then click Edit, then double-click Position.

4. Type 0.8, press Tab, type 4.7, press Tab, type 7.9, press Tab, and then click OK.

5. Repeat for point 2 with these coordinates (12.3, 3.9, 0.1).
6. Repeat for point 3 with these coordinates (24.1, 7.9, 7.9).

7. Drag the cursor over all three monitor points to select them.

8. Click Edit, click DRAG, click P, click P_IMP, and then click OK.

9. In the scenario menu click Single Field 3D output to expand it.

10. Click P, click PROD, click VVEC (this will highlight the three lines below VVEC).

11. In the scenario menu click Simulation and output control to expand it.

12. Double-click NPLOT, and type 50.

13. In the scenario menu click Gas composition and volume to expand it.

14. Double-click Fuel region, double-click Position, type 0, press Tab, type 0, press Tab, type 0.

15. Double-click Size, type 25.6, press Tab, type 8, press Tab, type 8.
2.6 Introductory example

Figure 2.15: Adding a gas cloud and choosing the gas composition

16. Click Edit volume fractions to open the volume fraction editor

17. Add METHANE and set volume fraction to 91.7

18. Add ETHANE and set volume fraction to 7

19. Add PROPANE and set volume fraction 1.3

20. Close the volume fraction editor by clicking OK

21. Double-click Equivalence ratios (scroll down to locate it), type 1.05, press Tab, type 0.

22. In the scenario menu click Ignition to expand it.


24. Click File, click Save, and click OK.

25. Minimise CASD.

2.6.4 Starting the FLACS-CFD simulation

Select the job in RunManager and click simulate (if the job is not visible then use add directory or if the directory is already added, right click and rescan); check how the simulation starts up (click log file).
2.6.5 Studying the results in Flowvis

2.6.5.1 Creating scalar time plots

1. In RunManager, click Tools, then click Flowvis.

2. Right-click in the page, then click Scalar Time.

3. In the Runs column find the directory and click 100001; in the Variables column click P, in the Monitors column shift-click to select all three monitors, then click OK.

4. In the properties sidebar enable the "Data range in legend" check-box.
2.6 Introductory example

5. Click in the pressure plot to activate it and open the Edit menu, choose Copy.

6. Click on the Page menu and choose Add page.

7. Open the Edit menu, choose Paste. You will now see the same plot as previously.

8. In the properties sidebar select DRAG as variable and deselect P.

9. Repeat the above steps to plot the P_IMP variable.

2.6.5.2 Creating a 2D contour plot

1. Open the Page menu, click Add.

2. Right-click in the plot area, choose 2D Cut Plane.

3. In the Runs column select 100001, in the Variables column click P_3D, then click OK.

4. In the properties sidebar, draw the Z slider to the value 4.5 m.

5. In the properties sidebar, open the Appearance menu and set Value Range to Fixed; fill in the range as 0.05 to 2.0.

6. Use the time slider at the bottom to move through the time steps.
2.6.5.3 Creating a single 2D contour plot with flame (PROD) and velocity (VVEC)

1. Open the Page menu, click Add.
2. Right-click in the plot area, choose 2D Cut Plane.
3. In the Runs column select 100001, in the Variables column click PROD_3D and VVEC_3D (use CTRL-click when combining the two variables), then click OK.
4. In the properties sidebar, draw the Z slider to the value 4.5 m.
5. Use the time slider at the bottom to move through the time steps.

2.6.5.4 Creating a scalar time plot of pressure (P) and a 2D contour plot of flame (PROD)

To create this plot on the same page as the previous one:

1. Click on the Page Subdivision button in the tool bar; select a layout with two plots (crosses) in the vertical, one in the horizontal.
2. Right-click in the lower half of the plot area, choose Scalar Time.
3. In the Runs column find the directory and click 100001; in the Variables column click P, in the Monitors column shift-click to select all three monitors, then click OK.
4. Use the time slider at the bottom to move through the time steps.

2.6.5.5 Creating a 3D contour plot with flame (PROD) and pressure (P)

1. Open the Page menu, click Add.
2. Right-click in the plot area, choose 3D Plot.
3. In the Runs column click 100001, in the Variable column click on P_3D and select Surface in the drop-down menu appearing next to it. Then CTRL-click on PROD (it should be set to Volume automatically). Click OK.
2.6 Introductory example

4. Use the mouse to rotate the geometry into a suitable angle.

5. In the properties sidebar, make sure that in the Appearance menu the variable is set to PROD. In the gradient drop-down menu select afmhot.

6. Use the time slider at the bottom to move through the time steps.

![3D plot with both surface (P) and volume (PROD) variables enabled.](image)

Figure 2.19: Creating a 3D plot with both surface (P) and volume (PROD) variables enabled.

2.6.6 Studying the effect of ignition location

1. In CASD, open 100001.caj, click file, then click Save As, move up a level, create a new directory called 100002, save the file as 100002.caj.

2. In the scenario menu, expand the Ignition section.

3. Click Ignition, then double-click Position, then type 0.5.

4. Click Porosities, then click Calculate.

5. Click file, then click save.

6. In RunManager, add the directory 100002, enable the check box next to the scenario and then click Simulate.

7. In RunManager, click Tools, then click Flowvis.


9. In the job number column click 100001, in the Variable column click P, in the Monitors column drag to select all three monitors, then click OK.

10. Right-click into the plot, enable “Data Range in Legend”.

11. Note the maximum value of pressure.

12. Save the file as 100002.fml.

13. Open the file 100001.fml, note the maximum value of pressure.
Chapter 3

CASD

CASD is an acronym for Computer Aided Scenario Design. The preprocessor CASD for the CFD simulator FLACS-CFD is used to prepare the input data, or job data, that defines a FLACS-CFD simulation. This comprises the: geometry model, computational grid, porosities, and scenario description.

This manual describes CASD 8, but the general functionality is in principle the same for earlier versions.

3.1 Overview

This section provides a general overview of the functionality in CASD, starting with an overview of how geometry is represented in FLACS-CFD.

3.1.1 Geometry in FLACS-CFD

Geometry is represented as a binary tree, with union and left difference operations between primitives and operations. The binary tree exists as an object that is part of a geometry file. When working with geometry in the Object window you will always see the geometry as a list of nodes (a node is an operation or primitive), this means that although the underlying representation is a binary tree, you will never see it as such.
3.1.1.1 Geometry representation

An example binary tree can be seen in the figure above. Operations are represented by circles, and primitives are represented by boxes. Primitives can have a separate representation known as the FLACS-CFD model, as illustrated by the stippled boxes in the figure. The standard model, i.e. not the FLACS-CFD model, is commonly referred to as the visualisation or CAD model.

The FLACS-CFD model can contain the following primitive types: boxes, cylinders, ellipsoids, CP8s and GTCs. These primitives will represent themselves in the FLACS-CFD model unless the FLACS-CFD model have been removed or replaced. An example of this can be seen in the figure above; the box will represent itself in the FLACS-CFD model. In the same example, the cylinder has an empty FLACS-CFD model and will therefore not exist in the FLACS-CFD model.

This allows you to have more advanced and visually pleasing geometry for presentation purposes, while insuring that the model used by the simulator only consists of simple geometric shapes that have been extensively validated. For mere visualisation purposes, it is also possible to have geometry in the visualisation model that has no FLACS-CFD representation, for example geometry to show an escape route.

The porosity calculator available in FLACS-CFD, FGC (Flacs Geometry Calculator), will default use the FLACS-CFD model if it is available, when calculating the porosities. When importing CAD models using geo2flacs the FLACS-CFD model is no longer generated, and the visualisation model will be used.

See also:

The section about the visualisation and FLACS-CFD model when importing geometry using geo2flacs.
The section about the object window and primitive list.

3.1.1.2 Primitives

The following primitive types are supported by FLACS-CFD:
• Box
• Cylinder
• Ellipsoid
• General truncated cone (GTC)
• Convex polyhedrons (CP8)
• Torus
• Rectangular torus
• Mesh

They all share a number of basic properties as seen in Common primitive properties.

Table 3.1: Common primitive properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the primitive. If name is set to &quot;Light&quot; the primitive is</td>
</tr>
<tr>
<td></td>
<td>interpreted as a light - see Light for more information.</td>
</tr>
<tr>
<td>Type</td>
<td>The type of primitive (e.g. box).</td>
</tr>
<tr>
<td>Material traits -&gt; Material name</td>
<td>Name of the material to be used. Will override anything set on the</td>
</tr>
<tr>
<td></td>
<td>database object level. When NONE is selected the primitive will not receive</td>
</tr>
<tr>
<td></td>
<td>any material regardless of what is specified on the database object level.</td>
</tr>
<tr>
<td></td>
<td>When DEFAULT is selected the primitive will inherit the material from the</td>
</tr>
<tr>
<td></td>
<td>first parent group that has a material specified; if no parent group has</td>
</tr>
<tr>
<td></td>
<td>a material specified the object will not receive any material.</td>
</tr>
<tr>
<td>Material traits -&gt; UV type</td>
<td>See UV types for more information.</td>
</tr>
<tr>
<td>Material traits -&gt; Drape direction</td>
<td>See Drape direction for more information.</td>
</tr>
<tr>
<td>Color</td>
<td>Defined by a hue, saturation and value (HSV) vector and a alpha value. The</td>
</tr>
<tr>
<td></td>
<td>hue must be in the range 0-360, saturation in the range 0-255 and the</td>
</tr>
<tr>
<td></td>
<td>alpha between 0-1.</td>
</tr>
<tr>
<td>Porosity</td>
<td>Area porosity as a vector and volume porosity as a float. 0 is closed,</td>
</tr>
<tr>
<td></td>
<td>1 is open. Expanding the porosity shows also the advanced option Trace</td>
</tr>
<tr>
<td></td>
<td>area porosity, used by FGC (Flacs Geometry Calculator). The default &quot;Auto&quot;</td>
</tr>
<tr>
<td></td>
<td>setting enables this option for large, zero thickness objects. It can be</td>
</tr>
<tr>
<td></td>
<td>forced to be on or off for the selected object.</td>
</tr>
<tr>
<td>Metadata</td>
<td>Can store arbitrary key-value data. Only supports strings.</td>
</tr>
<tr>
<td>Rotation</td>
<td>Rotation in degrees for each axis (x, y, z). The value is derived from</td>
</tr>
<tr>
<td></td>
<td>primitive specific properties, and is not editable.</td>
</tr>
</tbody>
</table>

3.1.1.2.1 Box  In addition to the properties specified in Common primitive properties, the box primitive supports the properties specified in Box primitive properties.

Table 3.2: Box primitive properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>Cartesian position in units</td>
</tr>
<tr>
<td>Size</td>
<td>x, y and z size in units</td>
</tr>
</tbody>
</table>
3.1.1.2.2 Cylinder  In addition to the properties specified in Common primitive properties, the cylinder primitive supports the properties specified in Cylinder primitive properties.

Table 3.3: Cylinder primitive properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>Cartesian position in units</td>
</tr>
<tr>
<td>Diameter</td>
<td>in units</td>
</tr>
<tr>
<td>Length</td>
<td>in units</td>
</tr>
<tr>
<td>Direction</td>
<td>One of X, Y and Z. Can be prefixed with a minus sign to set negative direction</td>
</tr>
</tbody>
</table>

3.1.1.2.3 Ellipsoid  In addition to the properties specified in Common primitive properties, the ellipsoid primitive supports the properties specified in Ellipsoid primitive properties.

Table 3.4: Ellipsoid primitive properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>Cartesian position in units</td>
</tr>
<tr>
<td>Size</td>
<td>in units</td>
</tr>
</tbody>
</table>

3.1.1.2.4 General truncated cone (GTC)  In addition to the properties specified in Common primitive properties, the GTC primitive supports the properties specified in GTC primitive properties.

Table 3.5: GTC primitive properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector 1 - Vector 6</td>
<td>Defines the primitive. See Definition of a general truncated cone for visual explanation.</td>
</tr>
<tr>
<td>Length</td>
<td>Derived from Vector 1 - Vector 6.</td>
</tr>
<tr>
<td>Diameter 1</td>
<td>Derived from Vector 1 - Vector 6, read only.</td>
</tr>
<tr>
<td>Diameter 2</td>
<td>Derived from Vector 1 - Vector 6, read only.</td>
</tr>
</tbody>
</table>

Figure 3.2: Definition of a general truncated cone.

3.1.1.2.5 Convex polyhedron (CP8)  In addition to the properties specified in Common primitive properties, the CP8 primitive supports the properties specified in CP8 primitive properties.

Table 3.6: CP8 primitive properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vector 1 - Vector 8</td>
<td>Defines the primitive. See Definition of a convex polyhedron for visual explanation.</td>
</tr>
</tbody>
</table>
3.1 Overview

3.1.1.2.6 Torus  A torus can be used to represent, for example, pipe bends. In addition to the properties specified in Common primitive properties, the torus primitive supports the properties specified in Torus primitive properties.

Table 3.7: Torus primitive properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theta</td>
<td>In radians. Pi=2 makes a closed tori.</td>
</tr>
<tr>
<td>Inner radius</td>
<td>Distance in units between the center of the tori and the inner wall.</td>
</tr>
<tr>
<td>Diameter</td>
<td>Diameter in units of the closed part of the tori. When combining a tori with a cylinder this should be set to the diameter of the cylinder.</td>
</tr>
</tbody>
</table>

3.1.1.2.7 Rectangular torus (rtorus)  A rtorus can for instance model a bent plate. In addition to the properties specified in Common primitive properties, the rtorus primitive supports the properties specified in RTorus primitive properties.

Table 3.8: RTorus primitive properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theta</td>
<td>In radians. Pi=2 makes a closed rtori.</td>
</tr>
<tr>
<td>Inner radius</td>
<td>Distance in units between the center of the rtori and the inner wall.</td>
</tr>
<tr>
<td>Diameter</td>
<td>Diameter in units of the closed part of the rtori in the major axis.</td>
</tr>
<tr>
<td>Height</td>
<td>Diameter in units of the closed part of the rtori in the minor axis.</td>
</tr>
</tbody>
</table>

3.1.1.2.8 Mesh  Mesh is a generic primitive that can take any shape. In addition to the properties specified in Common primitive properties, the mesh primitive supports the properties specified in Mesh primitive properties.

Table 3.9: Mesh primitive properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faces</td>
<td>Number of faces in the mesh. Uneditable</td>
</tr>
<tr>
<td>Vertices</td>
<td>The points the mesh consists of in units. Uneditable.</td>
</tr>
<tr>
<td>Is closed</td>
<td>When set to true the mesh is a closed 2-manifold. Uneditable.</td>
</tr>
<tr>
<td>Force closed</td>
<td>When set to true the mesh will be forcefully interpreted by the porosity calculator as being closed.</td>
</tr>
</tbody>
</table>

3.1.1.3 Operations

There are two different operation types supported by FLACS-CFD:
• Union

• Left difference

They share a number of basic properties:

Table 3.10: Common operation properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>The name of the operation</td>
</tr>
<tr>
<td>Type</td>
<td>The type of operation (e.g. union)</td>
</tr>
<tr>
<td>Material traits - &gt; Material name</td>
<td>Name of the material to be used. When NONE is selected the primitive will not receive any material. When DEFAULT is selected the primitive will inherit the material from the first parent group that has a material specified; if no parent group has a material specified the object will not receive any material.</td>
</tr>
<tr>
<td>Metadata</td>
<td>Can store arbitrary key-value data. Only supports strings.</td>
</tr>
</tbody>
</table>

3.1.1.3.1 Union  Named unions are also referred to as a group, and are visible to the user in the object window. Unnamed unions are hidden to the user, but are necessary to represent the geometry as a binary tree. Unions support the properties specified in Common operation properties.

3.1.1.3.2 Left difference  A left difference operation is a boolean difference operation applied between two nodes. The nodes can be any combination of operations or primitives supported by FLACS-CFD. The operation is named left difference because the geometry is represented in a binary tree, as shown in paragraph Geometry representation, and it is the right node in the tree that is subtracted from the left node.

See also:

See paragraph Left difference under subsection Menu bar in this chapter for information about how to create left difference operations. Advice on creating left difference operations is included in the Geometry modelling section of the FLACS-CFD Best practice chapter.

3.1.1.4 Materials

Table 3.11: Material properties

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Color</td>
<td>Specifies whether the primitive or material colour should be used. When the latter is selected it will override all colours specified on the primitive level. When a color texture is set and enabled its value will be blended with either the primitive color or material color, depending on the setting here.</td>
</tr>
<tr>
<td>Color texture</td>
<td>When set and enabled, the texture will be blended with the primitive or material color. Texture coordinates are defined on the primitive level - see Common primitive properties.</td>
</tr>
<tr>
<td>Normal map</td>
<td>When set and enabled this will modify the normal defined by the primitive so it can vary across a surface.</td>
</tr>
<tr>
<td>Roughness</td>
<td>Slider: Specifies the roughness/smoothness of a material. The slider generates a value between 0-1, where 1 means that the material is very rough. Roughness map: When set and enabled, the roughness value will be retrieved from the specified texture instead of the uniform value specified by the slider. Makes it possible to vary the roughness across a material surface.</td>
</tr>
</tbody>
</table>
3.1 Overview

<table>
<thead>
<tr>
<th>Property name</th>
<th>Description</th>
</tr>
</thead>
</table>
| Metallic      | *Dielectric*: When selected the material is non-metallic, i.e. the metallic value will be set to 0.  
*Metal*: When selected the material is completely metallic, i.e. the metallic value will be set to 1.  
*Metalness map*: When set and enabled the metallic value will be retrieved from the specified texture instead of a uniform value as with the other options. Makes it possible to vary the metalness across a material surface. This parameter only has an effect when Realistic lighting is enabled. |
| Scale         | The texture coordinates will be scaled according to the values specified here. |
| Rotate        | The texture coordinates will be rotated according to the values specified here. |
| Translate     | The texture coordinates will be translated according to the value specified here. |
| Wrap (u, v)   | Specifies how the texture lookup is done when the texture coordinates go outside the range 0-1, and can be set to one of the following:  
*REPEAT*: The texture will be repeated.  
*CLAMP*: Deprecated, should not be used.  
*CLAMP_TO_EDGE*: Values below 0 or above 1 will yield the texture edge value.  
*CLAMP_TO_BORDER*: Values below 0 or above 1 will result in a black color.  
*MIRROR*: The texture will be repeated, but be mirrored when the integer part of the texture coordinate is odd. |

3.1.1.4.1 **UV types**  See UV types for a complete list of the different UV types supported in FLACS-CFD.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Default. Same as <em>planar world</em>.</td>
</tr>
<tr>
<td>Custom</td>
<td>The primitive have custom texture coordinates. Only applicable to meshes.</td>
</tr>
<tr>
<td>Planar local</td>
<td>Planar texture coordinates in local space.</td>
</tr>
<tr>
<td>Planar world</td>
<td>Planar texture coordinates in world space.</td>
</tr>
<tr>
<td>Cylinder</td>
<td>Cylindrical texture coordinates.</td>
</tr>
<tr>
<td>Sphere</td>
<td>Spherical texture coordinates.</td>
</tr>
<tr>
<td>Drape</td>
<td>Planar texture coordinates in world space. UV direction is specified manually instead of automatically as with <em>planar world</em></td>
</tr>
<tr>
<td>Cube map</td>
<td>Cube map texture coordinates. Suitable for use with six sided textures.</td>
</tr>
<tr>
<td>Cubic</td>
<td>Cubic texture coordinates.</td>
</tr>
</tbody>
</table>

3.1.1.4.2 **Drape direction**  Only applicable in combination with the Drape UV type. Defines the direction in which the texture is draped.

3.1.1.5 **Light**

Lights can be defined in a FLACS-CFD geometry by naming a primitive "Light". Operations can not be used to define lights. Three different configuration parameters are supported by lights. The parameters must be added to the metadata property of the primitive. See Light configuration parameters for an explanation of the different parameters. Primitives can easily be converted to lights by using the functionality described in Convert. The lighting model supports only spherical lights and tube shaped lights, but any primitive can be converted to a light. For ellipsoids the spherical light model will be used, otherwise the tube shaped will be used, and fit to the primitives bounding box. Lights only have an effect when Realistic lighting is enabled.
Light calculations are both memory and computationally intensive, and adding too many lights to a geometry may cause a low frame rate and memory issues. Performance will depend heavily on the specification of your PC, and is affected by the available CPU power, memory size (RAM), and graphics card resources.

Table 3.13: Light configuration parameters

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AreaOfEffectRadius</td>
<td>Only nodes in a sphere defined by the light position and AreaOfEffectRadius</td>
</tr>
<tr>
<td></td>
<td>will be effected by the light. Having a large area of effect radius will</td>
</tr>
<tr>
<td></td>
<td>reduce performance.</td>
</tr>
<tr>
<td>EnableShadowCasting</td>
<td>When set to 0 the light will not cast shadows, all other values will enable</td>
</tr>
<tr>
<td></td>
<td>shadow casting. Enabling shadow casting will reduce performance.</td>
</tr>
<tr>
<td>LightIntensity</td>
<td>Defines the intensity of the light. The higher the value the more intense the</td>
</tr>
<tr>
<td></td>
<td>light will become.</td>
</tr>
</tbody>
</table>

3.1.1.6 Realistic lighting

Enabling realistic lighting changes the lighting computations to be more physically based, modelling Fresnel reflection and conservation of energy. It allows modelling of many different types of surfaces based on the two material properties, roughness and metalness.

The roughness parameter models surface irregularities not captured by the normal map, and can be set from 0% to 100%.

Metalness does not have a range, a material is either metal or non-metal, also called dielectric. The main difference between metal and non-metal materials is that metal reflects more light, and has no diffuse light reflection. Metal also has colored reflection, such as gold or copper, as some wavelengths of light may be absorbed.

Realistic lighting is not used when drawing semi-transparent objects.

Figure 3.4: Ellipsoids with roughness of 0%, 25%, 50%, 75% and 100%. Front row is metal and back row is non-metal.

3.1.1.6.1 Sky light When both realistic rendering and the sky is enabled, the sky will be used as the main light source. The position of the sun and color of the sky will affect the appearance of geometry.

3.1.1.6.2 Reflections Enabling reflections increases the realism and gives more spatial awareness by making surfaces reflect other surfaces around it. This is based on a screen-space reflection technique, computed after the geometry shown on the screen has been rendered. This means that surfaces can only reflect what is currently shown on the screen, so where it would normally reflect things outside the screen, or reflect obscured objects, the reflection will be incorrect. Reflections are computationally demanding and can be turned off.
3.1 Overview

3.1.1.6.3 Bloom  To emulate high dynamic range, colors above a certain brightness threshold, such as lights or some specular highlights, will appear to glow. Bloom is always enabled when realistic light is enabled.

3.1.2 System requirements

Please see the section on Hardware and software requirements regarding the graphics card requirements for running CASD.

In CASD, if insufficient OpenGL support is present, some aspects of 3D views may not work, for example the wireframe display. If updating the graphics driver does not help then a workaround to obtain wireframe views of 3D geometries is to start CASD with the Open Inventor-based viewer, see the following section.
3.1.3 Starting CASD

You start CASD by clicking the CASD icon in the RunManager window:

![The CASD desktop icon](image)

or alternatively by executing the command:

```bash
> run casd
```

on the command line in Linux. If CASD is unable to obtain a license when connecting to the FLACS-CFD license server, then an error message like “The application was unable to start correctly (0x0000142). Click OK to close the application.” will be shown. To resolve such a problem see the guidance about how to set up a network license or single user license.

3.1.4 CASD command line options

The following options can be given when starting CASD on the command line:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-macro file name</td>
<td>Read input from specified macro file</td>
</tr>
<tr>
<td>-noLock</td>
<td>Turns off locking on the database files. <em>Must not be used if more than one user accesses the database simultaneously.</em> This option speeds up the database operations significantly.</td>
</tr>
<tr>
<td>-viewer osg</td>
<td>Default: use the OSG viewer</td>
</tr>
<tr>
<td>-viewer iv</td>
<td>Use the Open Inventor-based viewer</td>
</tr>
<tr>
<td>-display and others</td>
<td>Linux: options accepted by X</td>
</tr>
</tbody>
</table>

Example:

Linux:

```bash
> run casd -noLock
```

Windows:

```bash
> casd -noLock
```

Alternatively, the options may be set permanently in the FLACS-CFD RunManager (click Options, then click Preferences). This will only apply if CASD is started from the RunManager.

3.1.5 The main window in CASD

Starting CASD 7 opens the main window.
3.1 Overview

Figure 3.8: The main window in CASD

The main window is divided into the following parts:

- The menu bar
- The tool bar
- The command input field
- The geometry window(s)
- The scenario menu
- The status field
- The status bar

These parts are described in the following subsections.

The geometry window and scenario menu are dockable and can be moved and resized within the CASD window by clicking in their top right corner or dragging them on the top border of their windows.

Attention:

When the dockable windows are in undocked state they may disappear when CASD looses focus (e.g. when activating a different program) and return only when CASD regains focus again. This behaviour is not a property of CASD but depends on the window manager and can be set in its options. In KDE, for example, launch `kcmshell4 kwinoptions` and then go to the Advanced tab and disable *Hide utility windows*. This setting can also be accessed via System Menu → Applications → Configure Desktop → Window Behavior → Advanced → Hide utility windows.

3.1.5.1 The menu bar

The menu bar contains the following menus:

- File
- Geometry
The options on the various menus are described in separate sections in this chapter.

3.1.5.2 The tool bar

The tool bar contains the following parts:

- Main section: provides shortcuts to several of the commands on the menu bar
  - New, Open, Save, Save as, Import, and Result on the File menu
  - Calculate and Verify porosities on the Porosities menu

- Graphics section: controls various features of the The geometry window
  - View splitting
  - Rectangle zoom
  - Spinning (toggle on/off)
  - Highlighting option, from filled only (0) to various degrees of contour highlighting (1-5)

- Utilities section: plan drawings and load filters
  - Add plan drawings as a design guide
  - Add load filters to load selected parts of a geometry

- Snap point section: Snap points are active points on objects and primitives
  - Snap to corner
  - Snap to edge
  - Snap to face

- Layout buttons to activate different layouts, and a Save layout button to create a layout based on the current arrangement of the CASD main window components. See also the menu entry Options - Layout manager.

Note that the sections of the tool bar can be in different order, depending on the active layout. You can define several layouts, which not only set the content of the tool bar, but also the arrangement of the other components in the CASD main window.
3.1.5.3 The command input field

The command input field represents an alternative interface between the user and CASD, in addition to the regular menus on the menu bar. The control input field contains a scrollable command history list, and a current command context indicator (left side). To control the command history list from the keyboard use:

- UP: retrieves the previous line from the command history list
- DOWN: retrieves the next line from the command history list
- RETURN: processes the content of the command input field

Hence, you can choose whether to use a menu option on the menu bar, for example: File → Exit → Yes (to exit and save) or to execute, after typing or retrieving, the following command in the command input field:

```plaintext
* file exit yes yes
```

Command line input will in many situations be the most efficient way to work with CASD. Other sections in this chapter present additional examples on how to use this feature.

Up to ten of the most recently used commands are stored after the end of a session.

**Examples: Using the command input field in CASD**

- Select a box primitive in an object. The following command moves the box to (2, 2, 2), and causes the properties dialog to be shown
  
  ```plaintext
  * edit properties 2 2
  ```

  - This is because the position is not completely specified. You do not need to specify all parameters, but you must include all values for the parameter specified.

- If you want to edit one of the last parameters in the dialog, it is not necessary to specify all the parameters in front. The parameter name can be used to indicate which parameter to edit
  
  ```plaintext
  * edit properties size 2 2 2 vol.por 0.5
  ```

- You can also supply the answer to a question in the input field. To delete an assembly/instance, CASD will ask to confirm the operation. To avoid the question dialog, type the following command
  
  ```plaintext
  * geometry delete yes
  ```

  - or shorter: * ge de y

- To direct the output from a list to a file, append the file name after the list command. For instance, to list geometries in the database, enter the following command, which will create the text file outfile.txt
  
  ```plaintext
  * geometry list outfile.txt
  ```

3.1.5.4 The geometry window

The geometry window in the main window displays the geometry, the computational grid and scenario parameters. In addition to the options on the View menu, there are several ways of manipulating the view:

- Rotation: left-click
- Panning: CTRL+left-click
- Zoom: rotate mouse wheel
- Rectangle zoom: right-click and select Rectangular Zoom
- Splitting and closing views: right-click+SELECT
3.1.5.5 The status field

The status field in the main window contains information concerning the: active database, project, geometry, grid, and units.

3.1.5.6 Geometry window icons

Each geometry window has its own icon bar underneath.

![Figure 3.9: Geometry window icons](image)

The geometry window icon bar has following the following functions.

- Switch between walk viewer and fly viewer
- Selection mode on
- View mode on
- Store camera position
- Re-store camera position
- Seek to picked point
- View all
- Switch between parallel and perspective projection
- Turn axis on/off
- View along X axis
- View along Y axis
- View along Z axis

3.1.5.7 Cursor coordinate field

A cursor coordinate field is located to the lower left of the geometry window. In Select mode this field shows the position of the mouse cursor in the geometry. In View mode this field shows the position of the current snap point, which is set by performing a CTRL+left-click in the geometry window. A right-click in the cursor coordinate field opens a context menu with options to copy the coordinate value to the clipboard, and to change the font in the field. Changes to the font, e.g. size or typeface, is saved in the user settings.
3.1 Overview

3.1.5.8 Status bar

The status bar in the main window display information concerning

- The current model being visualised (FLACS-CFD or visualisation model)
- If any load filter is enabled
- The current unit in use

3.1.6 Files in CASD

CASD stores job data in a set of files. For the arbitrary job number 010100, the most important files are the:

- **Header file**, `010100.caj`: ASCII file created by CASD; defines the cs, co, and cg files used by CASD.

- **Geometry file**, `co010100.geo`: binary file created by CASD; contains a list of primitives that define the geometry; used by FGC and Flowvis. The geometry file is also called the *obstruction file*, or *co-file*. It is not a direct input to the simulation, but it is used by FGC when generating the porosity file.

- **Grid file**, `cg010100.dat3`: binary file created by CASD; defines the computational mesh; used by CASD, Flacs, and Flowvis.

- **Porosity file**, `cp010100.dat3`: binary file created by the porosity calculator, FGC (typically from the Porosities menu in CASD); defines the porosities for each grid cell; used by Flacs and Flowvis.
• **Scenario file, cs010100.dat3**: ASCII file created by CASD; defines the general scenario (monitor points, output variables, fuel region, pressure relief panels, ignition position, etc.); used by CASD, Flacs, and Flowvis.

The **File menu** in the main window contains commands for creating, opening, and saving the various job files. See section **Files in FLACS-CFD** for further information.

### 3.1.7 Working with geometries in CASD

Implementing the geometry model in CASD is often the most time-consuming part of a project. For modern process facilities it may be possible to import the geometry from an existing CAD model, but for many installations the geometry must be constructed manually from drawings, photographs, etc.

A large project, such as a full probabilistic analysis, can involve hundreds of CFD simulations, and each simulation will typically produce 10 to 15 different files. Hence, it is important to organise the files in a well-structured manner.

The building blocks in a CASD geometry are instances of objects. The structure within an object is described in **Geometry representation**.

The list of information required to implement a typical process facility, such as an offshore oil platform or an onshore process plant, is quite extensive and includes:

- Plot plan
- Sectional drawings
- Piping plan
- HVAC layout
- Cable trays layout
- Framing plans
- Cladding
- Deck plan.

Most FLACS-CFD users find it convenient to define standardised axis directions, and the following convention is used by Gexcon for typical process facilities:

- East-West along the x-axis, with positive x towards the east.
- North-South along the y-axis, with positive y towards the north.
- Up-Down along the z-axis, with positive z pointing upwards.

This results in a conventional right handed coordinate system, where the lower south-western corner of the facility coincides with the origin (0,0,0).

Each object in a CASD geometry is assigned a material property, and each ‘material’ is assigned a colour hue from the $0 - 360^\circ$ colour circle. Many FLACS-CFD users assign certain hues to various structural elements, and the following convention is used by Gexcon for typical process facilities.

<table>
<thead>
<tr>
<th>Hue</th>
<th>Colour</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Red</td>
<td>solid walls and decks</td>
</tr>
<tr>
<td>30</td>
<td>Orange</td>
<td>pressure relief and louvred panels</td>
</tr>
<tr>
<td>60</td>
<td>Yellow</td>
<td>grated decks</td>
</tr>
<tr>
<td>120</td>
<td>Green</td>
<td>anticipated congestion</td>
</tr>
<tr>
<td>180</td>
<td>Cyan</td>
<td>equipment</td>
</tr>
</tbody>
</table>

Table 3.15: Colour convention used by Gexcon
A standardised colour scheme makes it more straightforward to review geometries from old projects.

### 3.1.8 Working with mesh type primitives

The traditional FLACS primitives like box and cylinder always represent a single solid object, but the mesh primitive can represent anything from a single surface or solid, multiple surfaces or objects, or part of an object or surface.

Mesh primitives are normally the result of importing CAD files when the shape could not be detected as any of the other primitive types.

For mesh primitives that represent solid objects, the orientation of normals on the surface of the mesh is used to determine what is the inside and outside of the mesh.

CASD will try to detect if the mesh is a closed solid, and this can be seen as the "Is closed" property on the mesh primitive in the object window. If the mesh primitive is known to represent a solid object, but is not detected as such, the "Force closed" option must be set. A mesh object that is not closed and not force closed, will not produce volume blockage!

In some cases, a single solid object will be represented as multiple mesh primitives. The menu option Merge mesh can then be used to merge the objects into one.

Conversely, a single mesh primitive can represent several objects. The menu option Split mesh or Convex decompose mesh can then be used to split the mesh into multiple mesh primitives.

Splitting a mesh may result in e.g. the inner surface of a hollow object being split into a separate mesh. This will then become a “inside out” mesh and should be removed.

### 3.1.9 Representing geometry: on-grid and sub-grid scale obstacles

To appropriately model the effect of obstacles, it is important that the obstacles are geometrically well-represented on the chosen grid. In most practical situations, it will not be possible to represent smaller obstacles on the grid, however these should still be included in the geometry model since they may be treated by proper sub-grid models. Larger obstacles like the floor or ground, the ceiling, the walls and larger equipment will be resolved on-grid. Such objects may be repositioned so that their edges fall on grid lines, see Geometry considerations for the grid.

The most challenging geometries to represent properly are repeated obstacles of the same size and spacing as the chosen grid spacing. In such cases, consider changing the grid to achieve a better representation; if this type of geometry is dominant, then the accuracy of the result depends on the representation of these objects being good enough. In cases where such objects are not dominant, one may pay less attention to how the geometry is represented. For normal offshore modules there will be a range of sub-grid obstacles which are more or less randomly distributed in space.

A sub-grid obstacle is an object with a size of less than 2 control volumes (grid cells); for such an object, sub-grid contributions to turbulence and flame folding are applied. Bigger obstacles are called on-grid.

In many experimental setups one will find repeated obstacles of the same size. The basic research on gas explosions has focused on the effect of obstacle arrays, rather than on the effect of more realistic geometries. Both categories are important in order to be able to validate tools like FLACS-CFD.
3.2 File menu

3.2.1 New

Shortcut: CTRL+N

Creates a new simulation job.

The New command in the File menu creates a new empty job. If there were unsaved changes to the current job, a dialog box is displayed, asking about saving the changes.

3.2.2 Open

Shortcut: CTRL+O

This command opens an existing set of simulation files. The default simulation files selection is defined in the `header file` (*.caj).

The Open command in the File menu opens an existing job.

If the file name is entered in the command input field, the path must be encapsulated in apostrophes, for instance:

`* open "../../Test/000000.caj"`

If the command is selected from the menu, or if no name is specified in the command input field, the Open dialog box is displayed, where the path and file name can be specified.

By default, the file filter is set for selecting CASD `header files` (*.caj). The file type filter can be changed to show `geometry files` (co*.dat3, co*.geo) or all files (*.*). CASD will then open all files with the same job number.

If there were unsaved changes to the current job, a dialog box is displayed, asking about saving the changes. CASD will display the contents of the geometry file in the graphic area after a file is successfully opened.

The contents of the geometry file can be edited using the Edit File command in the Geometry menu.

When opening a simulation whose `cs-file compatibility` field is different than the most current version you will be asked if you want to update the file to the latest version. If you decline to update the file, it will retain its original value when saved.

3.2.3 Save

Shortcut: CTRL+S

Saves the current simulation job (that is, the various files that define the job).

The Save command in the File menu saves the current job.

3.2.4 Save as

Shortcut: CTRL+SHIFT+S

The Save As command saves the current job under a new (user-defined) name (job number).

3.2.5 Import

Imports some specifications from another simulation job (for example, grid file, scenario file, etc.).

3.2.6 Export

Exports the visible geometry in CASD to file with the provided format (for example *.obj files).

3.2.7 Exit

Shortcut: CTRL+Q

Exits the CASD program.
3.3 Geometry menu

CASD stores the geometry in geometry files (.co-files). The commands in the Geometry menu in the main window, except the Edit File command, are available when connected to a database. The Save and Save as commands in the File menu writes the geometry to the geometry file.

3.3.1 Edit file

The Edit File command in the Geometry menu makes it possible to edit the geometry file (.co file) for the open job.

3.3.2 Statistics

The Statistics menu item allows to compute various properties of the geometry, such as surface area and volume. The values can be determined for specific parts of the geometry, and the results can be grouped into user-defined categories. Overlapping parts of the geometry will be identified and not be counted twice.

![Figure 3.11](image.png)

**Figure 3.11:** The geometry statistics tool in CASD.

**Note:**

The Statistics menu item opens a graphical front-end to the cofile2 tool. For the time being, the older cofile (deprecated) tool is still included in the FLACS-CFD package and can be only used from the command line. Please note that regions are defined differently in cofile and cofile2.

3.3.2.1 Prefilter

The prefilter option allows to specify what geometry to include in the calculation, e.g. by primitive type or hue. The filter is specified in the same way as the Quick filter used in the Object Dialog. Any geometry not matching the filter will be disregarded and not influence the results. An empty filter will include all geometry.
3.3.2.2 Regions

Regions can be set up to include or exclude parts of the geometry based on their location. Each region is an axis-aligned bounding box around the geometry to be included or excluded. The union of the regions to be included reduced by the union of the regions to be excluded defines the total region for which the geometry statistics will be prepared. For example an L-shaped region can be defined either as two include regions, or one include region and one exclude region. The result will be the same.

Double-click a region to edit its size. The region will be displayed as a bounding box in the CASD geometry that can be interactively positioned and resized.

3.3.2.3 Categories

Several categories and subcategories can be defined. Each category has a name and a filter rule. The name is a user-defined text, and the rule is a filter like the Quick filter used in the Object Dialog. The result values for a subcategory will be the sum of the result for all primitives matching the filter rule in the category and any parent category.

To define a top-level category, make sure that no category is selected in the list and click Add category. To define a subcategory, select the parent category and click Add category. Double-click the category name or filter rule to edit it.

The following subcategories have been predefined for cylinders:

<table>
<thead>
<tr>
<th>Name</th>
<th>Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class 4</td>
<td>size &gt;= 0 &lt; 2.5in</td>
</tr>
<tr>
<td>Class 3</td>
<td>size &gt;= 2.5in &lt; 5in</td>
</tr>
<tr>
<td>Class 2</td>
<td>size &gt;= 5in &lt; 9in</td>
</tr>
<tr>
<td>Class 1b</td>
<td>size &gt;= 9in &lt; 15in</td>
</tr>
<tr>
<td>Class 1a</td>
<td>size &gt;= 15in &lt; 31in</td>
</tr>
<tr>
<td>LARGE</td>
<td>size &gt;= 31in</td>
</tr>
</tbody>
</table>

3.3.2.4 Saving and loading setup

The prefilter, regions and category setup can be saved to a file or loaded from an existing one. Such a file can also be loaded by the cofile (deprecated) utility.

3.3.2.5 Counting

Depending on the size of the geometry, the size of the regions, and the complexity/number of overlaps in the geometry, the calculation can take some time. A progress bar is shown while the geometry is processed.

Note:

As a polygonal representation of the geometry may be used in the calculations, the resulting values may not be exact. In particular, cylinders that are

• overlapping other geometry, or
• are part of a left-difference operation, or
• are only partly within a counting region,

will be represented as an extruded hexadecagon when calculating the volume and area. The error is partly compensated, but deviations of a few per cent should be expected in such cases. Also, compensation is not feasible for all object combinations and operations.

When computing the length of primitives, the center line is used. Parts of the center line that intersect bigger primitives will not add to the length.

To view a report of the results, click the View report button. The results can also be saved to a text file.
3.3 Geometry menu

3.3.3 Import CAD file(s)

CASD allows to run a geometry import procedure using geo2flacs entirely from the graphical user interface. The figure below shows the import dialogue.

![Geo2flacs Dialogue](image)

Figure 3.12: Geometry import dialogue in CASD.

After adding the relevant files, the object type can be set in the drop-down menu for each file. Transformation parameters for rotation, translation and scaling can be filled in if necessary. These correspond to the geo2flacs options \(-r\), \(-t\), and \(-s\), respectively. Also filtering, grouping, and alignment options can be specified. The alternative to running the geometry import from the GUI is to use the geo2flacs utility for manual geometry import.

3.3.3.1 Advanced options

The Advanced options button brings up a dialogue with some additional options that can improve the import in some cases:

Advanced options for DWG/DGN import
Table 3.17: Geometry import advanced options for DWG/DGN import.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assume circles are connected</td>
<td>When enabled it may fix problems where pipes are not imported.</td>
</tr>
<tr>
<td>Assume all objects are solid</td>
<td>Disable this option if the import contains extra geometry that seems out of place. This is caused by the importer assuming that all objects are solid; however, in some cases there are helper lines or other objects in the CAD file that are not part of the actual geometry and should not be imported.</td>
</tr>
<tr>
<td>Group nodes at root level by group name</td>
<td>When enabled the primitives and operations will be placed in named groups at root level based on the DGN level/DWG layer each primitive belongs to. This is applied at object level.</td>
</tr>
<tr>
<td>Analyse DWG boundary representation to detect left difference</td>
<td>Enables experimental support for left difference objects found in DWG files. In some cases the feature will lead to poorer import of non left difference objects.</td>
</tr>
<tr>
<td>Import unrecognized objects as mesh</td>
<td>When disabled unrecognized objects will be imported as bounding boxes.</td>
</tr>
<tr>
<td>Merge non-solid shapes</td>
<td>When enabled the algorithm will try and merge primitives that together constitute a box, e.g. 6 planes/boxes that together form a box will be merged into a a single box.</td>
</tr>
<tr>
<td>Split lines into segments</td>
<td>When this option is enabled lines will be split into segments using intersecting faces</td>
</tr>
</tbody>
</table>

Advanced options for RVM import

Table 3.18: Geometry import advanced options for RVM import.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import ‘Insulation’ primitives</td>
<td>By default, the import will include geometry that is tagged as insulation in the CAD model, but this can be switched off here.</td>
</tr>
<tr>
<td>Import ‘Obstruction’ primitives</td>
<td>Normally, the import will disregard obstruction primitives, but these may be included by enabling this option.</td>
</tr>
</tbody>
</table>

Advanced options for geometry analysis

Table 3.19: Geometry import advanced options for geometry analysis.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate FLACS model (legacy)</td>
<td>Enabling this option will generate FLACS-CFD model primitives for mesh primitives on import.</td>
</tr>
<tr>
<td>Enable mesh curvature analysis</td>
<td>Enabling this option may improve the import, especially of pipes, if the CAD file contains many mesh objects (objects represented by triangles or polygons). In some cases the algorithm will misinterpret input and large unwanted objects may appear in the import. These can be removed manually, after which the result is normally much better than without this option enabled.</td>
</tr>
</tbody>
</table>
3.4 Object window

The object window opens when choosing the menu item Edit File in the Geometry menu.

The object window consists of 6 main GUI elements as seen in the image above. More information about each GUI element can be found below:

- Menu bar
- Tool bar
- Primitive list
- Properties widget
- Quickfilter

3.4.1 Menu bar

3.4.1.1 File menu

The entries in the File menu of the object window are explained below.

3.4.1.1.1 Save The changes are stored internally in CASD and will be written to the file when you use Save or Save As from the File menu in the main window. Closing the object window without saving will drop all unsaved changes.

3.4.1.2 Exit Before closing the object window, CASD asks whether the object should be saved.

3.4.1.2 Edit menu

The entries of the Edit menu in the object window are explained below.
Remarks:

To use the shortcuts displayed in this menu, the input focus must be on the 3D view; click \texttt{MOUSE+LEFT} in the 3D view to activate it.

3.4.1.2.1 Undo \quad \text{Shortcut: } \texttt{CTRL+Z}

The \textit{Undo} command allows the user to undo a large number of the commands available in the object window. The supported commands include, but are not limited to, the following

- Convert -> *
- Properties
- Translate
- Scale
- Add -> Box
- Add -> Cylinder
- ...

3.4.1.2.2 Redo \quad \text{Shortcut: } \texttt{CTRL+Y}

The \textit{Redo} command allows the user to redo a large number of the commands available in the object window. It supports the same commands as the \textit{Undo} command.

3.4.1.2.3 Convert \quad \text{The subcommands under } Convert \text{ allows the user to convert primitives to a box, cylinder, mesh, ellipsoid, terrain or light. If the primitive cannot be converted an error message will be displayed.}

See also:

\texttt{Light} for the specification of a light.

3.4.1.2.4 Mesh tools \quad \text{The mesh tools menu is used when working with mesh type primitives.}

- Split mesh - Split the mesh into multiple disconnected components.
- Merge mesh - Opposite of split mesh, combines the selected meshes into one mesh.
- Repair mesh - Fixes small holes, degenerate faces or singular vertices in the mesh.
- Convex decompose mesh - Split the mesh into convex parts.
- Make normals consistent - Reorient normals so they point out of the mesh in a consistent manner.

3.4.1.2.5 Auto-align geometry \quad \text{This will check alignment of the selected primitives against X- and Y-axes, and if not determined to be aligned, the entire geometry will be rotated around Z-axis so that the selected primitives become aligned. If already determined to be aligned, nothing will be done.}

3.4.1.2.6 Properties (deprecated) \quad \text{Shortcut: } \texttt{P}

The \textit{Properties} command in the Edit menu changes the primitive properties if the selected node is a primitive. If you have selected a subtree containing only one type of primitives, the \textit{Properties} command can be used for changing one or more parameters for all these primitives.

3.4.1.2.7 Translate \quad \text{Shortcut: } \texttt{T}

Use the Translate command to move the selected subtree a specified distance per axis direction.
3.4 Object window

3.4.1.2.8 From To  Shortcut: F
Use the From To command to translate a subtree so that one specified position, the base point, is moved to another position, the target point. A dialog box for specifying the two positions is displayed. A circle is displayed in the 3D view, indication the position being edited. CASD keeps a list of positions used in the object. By pressing CTRL++ or CTRL+-, you can parse this list. The coordinates in the dialog box are updated.

3.4.1.2.9 Rotate  Shortcut: R
The Rotate command rotates the selected subtree. You must specify a base point for the rotation, the axis, and the rotation angle. As for the From To command, you can parse the position list using the CTRL++ or CTRL+- commands. If the rotation angle is not a multiple of 90 degrees then CASD will warn about creating objects that are not grid-aligned.

3.4.1.2.10 Scale  The Scale command scales the selected subtree. The origin for the scaling, and the scaling factor must be set. The position list can be parsed using the CTRL++ or CTRL+- commands before selecting Scale. The origin can be set by clicking CTRL+MOUSE+LEFT.

3.4.1.2.11 Scale XYZ  Shortcut: SHIFT+S
The Scale XYZ command scales the selected subtree independently in each axis direction. The origin for the scaling, and the scaling factor must be set. The position list can be parsed using the CTRL++ or CTRL+- commands before selecting Scale XYZ.

3.4.1.2.12 Delete subtree  Shortcut: Delete
The delete subtree command deletes all selected nodes in the primitive list and their children.

3.4.1.2.13 Delete node  Shortcut: Backspace
The delete node command deletes all selected nodes in the primitive list but will not delete any children unless they are explicitly selected.

3.4.1.2.14 Mark  Shortcut: CTRL+M
The Mark command is used when creating Left difference objects. It is possible to mark one or more nodes. Marked nodes are used as the positive part when adding a left difference operation. In order to mark nodes the user must select one or more nodes and trigger the Mark command.

See also:

The Advanced left difference operations section for example usage of the Mark command.

3.4.1.2.15 Duplicate  Shortcut: D
The Duplicate command in the Edit menu duplicates the selected primitives and operations. You are asked to enter the number of copies, and the distance between each copy in the three axis directions.

See also:

The Duplicate command in the Geometry menu duplicates the selected instance.

Creating pipe bundles
Start with creating one cylinder with the appropriate diameter, length and direction. Use the Duplicate command in the Edit menu to duplicate the cylinder in one direction. Use the same command once more to duplicate the resulting row of cylinders in the other direction.
If you need to change some parameters for all the cylinders, select all the primitives and use the Properties command. If you want to change the distances between the cylinders, this can be done by scaling all the primitives. Afterwards you can use the Properties command to reset the cylinder diameters and lengths.
3.4.1.2.16 Material Shortcut: M
The Material command in the Edit menu edits the material name for the object. You must enter an existing material name. This action is only enabled when connected to a geometry database.

3.4.1.2.17 Open material editor Shortcut: Shift+N
Opens the "Edit materials" dialog that shows all materials available to the object or geometry.

3.4.1.2.18 Open selected material Shortcut: Ctrl+Shift+C
Opens the material specified for the selected primitive for editing.

3.4.1.2.19 Change color This command allows the user to change the color of primitives and operations. The dialog that is opened by this command also allows the user to set the transparency (Alpha channel). The color can be input as HSV, RGB or through the color picker.

3.4.1.2.20 Copy Shortcut: CTRL+C
This function copies the selected primitives and operations to the system clipboard.

3.4.1.2.21 Cut Shortcut: CTRL+X
This function cuts the selected primitives and operations and places them on the system clipboard.

3.4.1.2.22 Paste Shortcut: CTRL+V
This function pastes primitives and operations that was previously stored in the clipboard by copy or cut.

3.4.1.2.23 Remove FLACS-CFD geometry This will remove the FLACS-CFD representation, and have the same effect as deleting primitives from the FLACS-CFD model.

3.4.1.2.24 Reset FLACS-CFD geometry This will reset the FLACS-CFD representation back to the default representation of the selected primitives. If for example the FLACS-CFD representation has been removed, this can be used to bring it back. For Mesh primitives, you can choose to use Mesh Curvature Analysis when creating the FLACS-CFD primitives.

3.4.1.2.25 Replace geometry with FLACS-CFD geometry This will replace the selected visualisation primitives, with their corresponding FLACS-CFD representation, or remove them if the FLACS-CFD representation is empty.

3.4.1.2.26 Edit FLACS-CFD geometry The action (shortcut F4) can be used to switch between editing the visualisation model and the FLACS-CFD model. This will show the two models side by side, and the view currently being edited will be highlighted.
When the view is changed the selection in the current view will be used to set the initial selection in the new view.
In most cases it should be sufficient to edit the visualisation model. Any additions or removals to the visualisation model will also affect the FLACS-CFD model. Removing nodes from the FLACS-CFD model will only affect the FLACS-CFD model and not the visualisation model.
To add Primitives in the FLACS-CFD model the user must be editing the FLACS-CFD model, and a single node must be selected in the primitive list. Anything added will then become part of the FLACS-CFD representation of the currently selected visualisation primitive.

3.4.1.2.27 Group Creates a group (named union) containing the selected primitives and operations. This will prompt the user for a group name, and put the selected items under a new top-level group.
3.4 Object window

3.4.1.2.28 Ungroup  Removes the selected group or the parent group of the selected primitives. The primitives in the group will not be removed.

3.4.1.2.29 Create instance...  Selected primitives will be converted to an instance object, which then can be duplicated with minimal overhead. This is useful if the geometry include a large number of identical objects, e.g., trees in a forest, as it will speed up drawing, reduce memory consumption and reduce the size of the geometry file.

3.4.1.2.30 Replace instance with geometry  The selected instance will be converted to a normal primitive, i.e. made unique.

3.4.1.2.31 Edit instance  Opens a new object editor window where a selected instance object can be edited. This will update all the duplicates of the instance object in the geometry model.

3.4.1.2.32 Export to terrain
Allows the user to export all nodes to a terrain .tri file. See Terrain import in CASD for more information about terrains.

3.4.1.3 Add menu
The Add menu allows to add primitives to the object. The different primitives and operations are described below.

Remarks:
To use the shortcuts displayed in this menu the input focus must be on the 3D view; click MOUSE+LEFT in the 3D view to activate it.

3.4.1.3.1 Box  Shortcut: B
The Box command in the Add menu adds a box to the end of the primitive list. A dialog box for defining the box parameters is displayed. See Box for a definition of the parameters.

3.4.1.3.2 Cylinder  Shortcut: C
The Cylinder command in the Add menu adds a cylinder to the end of the primitive list. A dialog box for defining the cylinder parameters is displayed. See Cylinder for a definition of the parameters.

3.4.1.3.3 Ellipsoid  Shortcut: E
The Ellipsoid command adds an ellipsoid to the end of the primitive list. A dialog box for defining the ellipsoid parameters is displayed. See Ellipsoid for a definition of the parameters.

3.4.1.3.4 CP8  The CP8 command adds a convex polyhedron to the end of the primitive list. A dialog box for defining the convex polyhedron parameters is displayed. See Convex polyhedron (CP8) for a definition of the parameters.

3.4.1.3.5 GTC  The GTC command adds a general truncated cone to the end of the primitive list. A dialog box for defining the general truncated cone parameters is displayed. See General truncated cone (GTC) for a specification of the primitive type.

3.4.1.3.6 Torus  The Torus command adds a torus to the primitive list. See Torus for a specification of the primitive type.

3.4.1.3.7 Rectangular torus  The Rectangular Torus commands adds a rectangular torus to the primitive list. See Rectangular torus (rtorus) for a specification of the primitive type.
3.4.1.3.8 **Left difference**  Shortcut: L
The Left Difference command adds a difference operation at the end of the primitive list. The marked nodes (Mark) are treated as the solid part of the operation, while the selected nodes are treated as the minus part. Both the solid and minus part of the left difference operation can be a collection of primitives and/or operations. When more than one node is marked as the solid part of the operation a named union will be created. A named union will also be created if more than one node is selected as the minus part of the operation.

**Note:**

One full grid cell is required inside the solid walls around “left difference holes” to ensure that the walls do not leak. In case of “minus cylinders” inside a solid cylinder, at least two CVs are needed, whereas one CV is sufficient for minus cylinders inside a box.

**See also:**

Advice on creating left difference operations is included in the Geometry modelling section of the FLACS-CFD Best practice chapter. A specification of the left difference operation can be seen in the Left difference paragraph.

3.4.1.3.9 **Copy**  Shortcut: SHIFT+C
The Copy command adds a copy of the selected primitives and operations.

3.4.1.3.10 **Object**  Shortcut: O
The Object command adds a copy of a specified object.

3.4.1.3.11 **From file**  Adds primitives and operations from a geometry file.

3.4.1.4 **Select menu**

There are several ways to select multiple primitives in an object window, and these are accessible in the Select menu. The entries of the Select menu are explained below. Later, it will also be explained how to select primitives with the mouse.

3.4.1.4.1 **Bounding box**  Shortcut: CTRL+B
This command lets you define a bounding box, and all primitives within it will be selected. If subsequently using the Cut or Copy operation, primitives intersecting the bounding box will be cut or copied at the intersection plane. For example the Cut operation can be used to cut a portion out of a Box primitive when it is partly overlapped by a bounding box.

3.4.1.4.2 **Filter**  This command will set up a filter in the same way as the load filter in the main geometry view. Any primitives matching the filter rules will be selected.

3.4.1.4.3 **All**  Shortcut: Ctrl+A
Select all primitives.

3.4.1.4.4 **Invert**  Shortcut: CTRL+I
Selects all primitives not currently selected.

3.4.1.4.5 **Stack**  Shortcut: CTRL+TAB
This command will parse (cycle through) the list of selected primitives or subtrees if more than one is selected.
3.4 Object window

3.4.1.4.6 Group  Shortcut: CTRL+|
Will select the first named group of the selected primitive or operation.

3.4.1.5 View menu
The entries of the View menu in the object window are explained below.

3.4.1.5.1 The XY, XZ and YZ views  Shortcut: Z
Shortcut: SHIFT+Z
Shortcut: Y
Shortcut: SHIFT+Y
Shortcut: X
Shortcut: SHIFT+X
The option XY View and XZ View display a projection of the geometry in the XY and XZ planes respectively.
The options YZ East View and YZ West View display a projection of the geometry in the YZ plane along
the positive and negative Y-axis respectively.

3.4.1.5.2 3D View  Shortcut: 3
The 3D View option displays a default 3D view of the geometry.

3.4.1.5.3 Maximize  Shortcut: SHIFT+M
The option Maximise maximises the visible window to display the entire geometry.

3.4.1.5.4 Axis  Shortcut: A
The Axis option turns axis display on and off.

3.4.1.5.5 Terrain  Toggle on/off terrain visualisation.

3.4.1.5.6 Grid Display  This function has no effect in the object window.

3.4.1.5.7 Annotation  This function has no effect in the object window.

3.4.1.5.8 Draw Style  Different options are available in this menu:
• Off: The geometry will not be displayed.
• Default: Geometry is drawn according to database/primitive settings.
• Opaque: All geometry will is drawn opaque.
• Semi-transparent: All geometry is drawn semi-transparent.
• Wireframe: Only the edges of the objects will be drawn.
• Wireframe and default: Combination of above descriptions.
• Wireframe and opaque: Combination of above descriptions.
• Wireframe and semi-transparent: Combination of above descriptions.
• Realistic: Enable realistic lighting.
• Use textures: Enable use of textures and normal/roughness/metalness maps on geometry.
• Use reflections: Enable reflections (only available when realistic lighting is enabled).
• Use shadows: Enable shadows (only available when realistic lighting is enabled).
• Show sky: Show skydome and use sky light instead of head light if realistic lighting is enabled.
3.4.1.5.9  Fixed vertical axis  When turned on it is not possible to rotate across +Z.

3.4.1.5.10  Fixed near plane  In some situations parts of the geometry may disappear when they are very close to the camera. Toggling this function will fix this problem.

3.4.1.5.11  Hide/Unhide  The actions in this menu makes it possible to hide one or more primitives from the 3D view. When hiding a primitive it will become grayed out in the Primitive list.

See also:

Please note there is also a global option to toggle on/off the show/hide functionality. See Toggle Hide Objects for more information.

3.4.1.5.12  Flacs view  Enables and disables the Flacs view window.

3.4.1.6  Macro menu

The entries of the Macro menu in the object window are explained below. For more information about CASD macros see section Macro menu (deprecated).

3.4.1.6.1  Run  Read a macro file defining a single object.

Attention:

Macro files generated by Write Object have to be read using the Run command from the Macro menu in the CASD main window.

3.4.1.6.2  Record  Writes all commands given to a user-specified macro file.

3.4.1.6.3  Write Object  Writes a macro file containing the current object.

Attention:

Macro files generated by Write Object have to be read using the Run command from the Macro menu in the CASD main window.

3.4.2  Tool bar

The available actions are in large parts identical to those described in section CASD - The main window in CASD - The tool bar. The actions that are only available in the object window is described below.

3.4.2.1  Toggle Hide Objects

Shortcut: F8
When triggered this will enable/disable show and hide functionality for this object window.

3.4.3  Primitive list

The primitive list shows all primitives in an object. It will show the primitive name and the type for each primitive. Groups are shown with bold text. Primitives that are shown in italic text are the FLACS-CFD model; see Geometry in FLACS-CFD for more information about the difference. Primitives that have an asterisk behind the name has an empty FLACS-CFD model.
3.4.4 Properties widget

Allows you to quickly edit properties for the selected primitives and operations. When multiple primitives and operations are selected only intersecting properties will be shown. This feature replaces the deprecated Properties dialog.

For an overview of the different properties that can be set please see section Geometry in FLACS-CFD of this chapter.

3.4.5 Quickfilter

This command will set up a filter for selection, but defined by a string instead of the user interface as in the filter command. The available filters are:

<table>
<thead>
<tr>
<th>Expression</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>box</td>
<td>Matches the corresponding primitive type.</td>
</tr>
<tr>
<td>cylinder</td>
<td>Matches if the specified dimension compared with the SIZE using the specified comparison operator is true. For boxes, length, width and height are defined as the longest, median and shortest dimension, respectively. For cylinders, length is always the cylinder length and width and height are equal to the diameter. It is possible to combine the dimensions into an expression, for example: width/height &gt; 2 would match primitives where the ratio width/height is greater than two.</td>
</tr>
<tr>
<td>gtc</td>
<td></td>
</tr>
<tr>
<td>cp8</td>
<td></td>
</tr>
<tr>
<td>ellipsoid</td>
<td></td>
</tr>
<tr>
<td>width &gt;</td>
<td></td>
</tr>
<tr>
<td>height &lt;</td>
<td></td>
</tr>
<tr>
<td>length = SIZE</td>
<td>Matches if the primitive hue is equal to the specified HUE.</td>
</tr>
<tr>
<td>diameter &gt;=</td>
<td></td>
</tr>
<tr>
<td>&lt;=</td>
<td></td>
</tr>
<tr>
<td>hue = HUE</td>
<td></td>
</tr>
<tr>
<td>aligned</td>
<td>Matches if the primitive is axis-aligned.</td>
</tr>
<tr>
<td>direction x or y or z</td>
<td>Matches if the primitive direction matches the specified direction.</td>
</tr>
</tbody>
</table>

The filter types can be combined using and and or operators and inverted using the not operator. For example:

```
box and (hue = 120 or length > 2)
```

selects boxes that are green or have length greater than two,

```
cylinder and not aligned
```

selects cylinders that are not axis-aligned.

A GTC is considered (axis-)aligned if its direction vector points along one of the coordinate axes. A CP8 is axis-aligned if three of the vectors are axis-aligned (those that span the origin corner if the CP8 were a box).

The direction of a cylinder is given by its axis, a GTC’s direction is given by the direction vector, and for a box or ellipsoid the length dimension (the one with greatest extent) determines the direction; if the two maximum extents of a box or ellipsoid are equal then no direction will be defined. Also for a CP8 the direction is undefined and will not match direction x/y/z.

3.4.6 3D view

3.4.6.1 Selecting primitives with the mouse

Besides the methods for selecting primitives that were explained above, it is also possible to use the mouse. To select primitives, Selection mode has to be enabled; it has the same icon as is used in the
geometry window. Left-clicking on a primitive will select it. By holding down CTRL and left-clicking several primitives successively, you can select multiple primitives. If you want to select a group of primitives that can be framed by a rectangle, you can press SHIFT on the keyboard and hold down the left mouse button while drawing a frame around the group of primitives that you want to select. See the image below for an example. When using this technique for selecting groups of primitives, it is important that there are no other primitives in front or behind of the intended group within the frame, otherwise these will be selected as well.

Figure 3.14: Multi-selection of primitives. Left: create a selection window with the mouse while holding the SHIFT key; right: all primitives in the selection window are selected.

Note:

The same ways of selecting primitives with the mouse also work in the geometry window.

3.5 Grid menu

The simulation volume is divided into grid cells (control volumes) by three sets of grid planes, one in each axis direction.

There is always a current grid working direction, and a selected region of grid planes in this direction. The current working direction is shown in the message area. The lines indicating the selected region are highlighted in the viewer.

3.5.1 Quick grid

Shortcut: CTRL+G

Using the Quick grid setup, it is possible to quickly create a grid with a core region covered by cells of a uniform size, surrounded by a stretched region where grid cells may vary in size.
3.5 Grid menu

The Quick grid setup window is divided in 4 sections:

- the boundaries and cell sizes for the core region are defined in metres in the first two sections,
- the stretched region of the grid is defined in metres in the third section, and
- information about the generated grid is shown in the fourth section.

Most parameters in the Quick grid dialogue require three values, associated with the X, Y and Z axes in that order. Values can be entered using the keyboard, the mouse scroll wheel, or by clicking the arrow buttons on the input field.

The values for the three axes are independent unless stated otherwise. If a relationship between two or more parameters is defined and no axis is specified, then the described relationship applies to each axis. For example, the expression

\[(\text{Cell size}) \times (\text{Number of cells}) = (\text{Actual maximum})\]

should be interpreted as

\[(\text{Cell size } X) \times (\text{Number of cells } X) = (\text{Actual maximum } X)\]
\[(\text{Cell size } Y) \times (\text{Number of cells } Y) = (\text{Actual maximum } Y)\]
\[(\text{Cell size } Z) \times (\text{Number of cells } Z) = (\text{Actual maximum } Z)\]

Note that it is possible to enter combinations of values into the Quick grid dialogue which cannot be used to define a grid that satisfies the Quick grid requirements. When this happens, an error message is displayed, and the field that should be adjusted is highlighted in red. Adjusting the stretched domain usually solves this.

When a scenario is saved and a grid is generated using the quick grid setup, the configuration parameters for the quick grid are saved in a “quickGrid{jobnumber}.json” file. When the scenario is reopened in CASD, this settings file is used to populate the quick grid setup. If a scenario with no associated quick grid settings file is opened in CASD, then CASD will try to set appropriate quick grid settings. These settings may not be appropriate and should be checked.

See also:

- Recommended grid configuration provides recommendations for defining grids appropriate for specific scenarios.

Below follows a short description of the various parameters.
3.5.1.1 Core domain

**Minimum**  The location where the core domain starts in this dimension.

**Maximum**  The highest coordinate for the core domain in this dimension.

**Actual maximum**  The location where the core grid actually ends; it is given by the smallest multiple of Cell size that is larger than or equal to Core domain - Maximum.

**Uniform grid**  If checked, the value of Cell size for the Y and Z axes is set equal to the value for the X axis. Same for the Max factor and Max cell size parameters.

**Cell size**  If selected, the cells in the core domain are guaranteed to have exactly the specified size. The value of Number of cells is computed automatically.

**Number of cells**  If selected, the core domain is guaranteed to have exactly the specified number of cells. The value of Cell size is computed automatically. If Uniform grid is checked, the Cell size for the X axis is computed automatically, and the value is used to determine the Number of cells for the Y and Z axes.

3.5.1.2 Stretched domain

**Stretch domain**  If checked, the core domain is surrounded by a stretched domain. The size and the number of cells in the stretched domain are computed automatically. The size of the cells in the stretched domain is guaranteed to be larger than or equal to Cell size.

**Minimum**  The location where the stretched grid starts. Must be less than or equal to Core domain - Minimum. If equal to Core domain - Minimum, no stretched domain is generated on the low side of the core domain.

**Maximum**  The location where the stretched grid ends. Must be greater than or equal to Core Domain - Maximum. If less than or equal to Core domain - Actual maximum, no stretched domain is generated on the high side of the core domain.

**Max factor**  The ratio between the size of two neighbouring cells is guaranteed to be less than or equal to this value. If Uniform grid is checked, the Max factor is the same for all the three axes. Otherwise, a specific Max factor can be defined for each axis. Only values between 1.0 and 2.0 are allowed. Suggested setting: 1.2.

**Max cell size**  If enabled, the size of a cell is guaranteed to be less than or equal to the specified value. If Uniform grid is checked, the Max cell size is the same for all three axes. Otherwise, a specific Max cell size can be defined for each axis. This value must be larger than or equal to Core domain - Cell size.

See also: Grid domains for more information about core domains and stretched domains.

3.5.1.3 Grid information

**Memory consumption**  An estimate of the amount of memory required by FLACS-CFD to calculate a simulation on the resulting grid.

**Core aspect ratio**  The ratio between the largest and the smallest dimension of cells in the core domain. If Uniform grid is checked, the core aspect ratio should be exactly 1.0. The core aspect ratio should never exceed 100, and a value of 1.0 is recommended for explosions, see the recommendations for configuring the grid.
Min cell  The size of the smallest cell in the grid.

Max cell  The size of the largest cell in the grid. If Max cell size is enabled, this value is guaranteed to be smaller than or equal to Max cell size.

Actual max factor  The actual maximum ratio between the size of two neighbouring cells. This value is guaranteed to be smaller than or equal to Max factor.

3.5.2 Check Grid

Shortcut: CTRL+SHIFT+G

The Check Grid menu choice opens a tool that can help to evaluate the quality of the current grid by checking it against recommendations in the grid guidelines. Please be aware that scenarios that include many scenario items might result in conflicting guidelines, which means that not all the recommendations can be fulfilled.

Note:

The evaluation does not consider the scenario geometry which might require additional recommendations according to the guidelines.

In addition to information about scenario type and items (leaks, panels, monitor points, fuel regions, etc.) read by the tool from the scenario setup, it is possible for the user to define a region of interest and confinement (i.e., if the scenario is confined by geometry on any sides). In addition, the user can set the ground level to exclude recommendations in negative Z direction that are not relevant as the grid is normally not extended below the ground level.

Based on the input from the scenario and the user, the tool will define a set of recommendations based on the grid guidelines, and test these recommendations against the current grid. The results are listed in a table as shown in the image below.

Figure 3.16: Example of the grid check dialog with recommendations for a simple fire scenario with a single jet leak. The white box in the 3D View represent the domain area related to the recommendation highlighted in blue.
Note:

The scenario needs to include a fuel region, a jet leak or a pool for the grid checker to be able to compute a core domain needed for generating a set of recommendations.

The columns in the table contains information related to each recommendation:

**Status:** Indicates if a recommendation is broken or OK.

**Source:** Indicates what triggered the inclusion of the recommendation (e.g. a specific scenario item, or if this is a general recommendation).

**Guideline:** Specifies the recommendation, e.g., that the number of cells in a defined area should be higher than a given threshold value.

**Threshold:** Specifies the threshold for the recommendation.

**Domain:** Specifies the target area for the recommendation, e.g. a range for a specific grid axis, an area or a volume. When clicking on a row, the target area is visualized in the 3D view as a white line, rectangle or box, depending on the shape of the target area.

**Result:** If the recommendation failed, this column will show the current value so that it can be compared to the threshold value.

Note:

The check grid tool is provided to help the user setting up a grid that follows the grid guidelines, but although a grid does not fail any recommendations in the grid check dialog this does not guarantee that the grid follows all the grid guidelines. It is the user's responsibility to make sure that a grid is following the guidelines for a given scenario.

### 3.5.3 Simulation volume

The *Simulation volume* command allows adjustment of the extent of the simulation volume in the three coordinate directions. If the volume is increased, the original grid planes are kept and one additional plane is added in each direction. If the volume is decreased, planes outside the new volume are deleted and new planes are created at the volume borders.

### 3.5.4 Direction

The Direction command changes the working direction. Legal inputs are *x*, *y* or *z*. The Grid menu commands Region, Add, Position, Move, Delete, Smooth, Stretch and List are implemented for grid planes in the working direction.

### 3.5.5 Region

The Region command substitutes the selected grid planes with a new set of grid planes and requires the new number of control volumes in the region to be entered.

### 3.5.6 Add

The Add command adds a new grid plane in the working direction and requires the coordinate value for the new plane to be entered.

### 3.5.7 Position

The Position command allows the position for the selected grid plane to be edited.
3.5 Grid menu

3.5.8 Move
The Move command moves the selected grid plane(s) a specified distance.

3.5.9 Delete
The Delete command deletes the selected grid plane(s).

3.5.10 Smooth
The Smooth command substitutes the selected grid planes with a new set of grid planes.
When the Smooth command is implemented, the sizes of the control volumes at each end of the region remain unchanged and sizes of control volumes between them is varied gradually, see the example in Grid guidelines. This function is typically used when refining the grid around a leak.

3.5.11 Stretch
The Stretch commands substitutes the selected grid planes with a new set of grid planes. This is particularly useful when stretching the grid towards the domain boundaries.
The Stretch command has two sub-choices:
- Negative Direction (typically used at the boundaries at the negative end of the axis)
- Positive Direction (typically used at the boundaries at the positive end of the axis)
The size of the control volume at one end of the region is required (the default is the current size). The factor by which the sizes of the control volumes in the specified direction should be increased or decreased is also required.

Attention:
Stretching of the grid should be avoided in areas of the simulation domain where the main combustion takes place. The flame model in FLACS-CFD in areas where accurate results are required. It is, however, prudent to stretch the grid towards the domain boundaries to reduce simulation time and computer memory requirements.

3.5.12 Information
The Information command displays status information about the defined grid, while the List command lists the grid coordinates in the working direction.

3.5.13 List
The Information command displays status information about the defined grid, while the List command lists the grid coordinates in the working direction.

3.5.14 Display
Three mutually exclusive options are available in the Display menu:
- Off: The grid is not displayed. Only the geometry is shown.
- Working Direction: The grid is displayed in the working direction only.
- All Directions: The grid is displayed in all three directions.
3.5.15 Select
The selected region includes grid planes between the planes that are selected as the lower and upper limits.
If only one plane is selected, the upper and lower limits are the same grid plane. Grid planes are selected
using the following commands:

- **Lower boundary**
  - Select the next grid plane: CTRL+RIGHT
  - Select the previous grid plane: CTRL+LEFT

- **Upper boundary**
  - Select the next grid plane: CTRL+UP
  - Select the previous grid plane: CTRL+DOWN

3.5.16 Grid-related operations

3.5.17 Importing the grid from another job
Use the Import command in the File menu to import the grid from another job. If a grid file name is entered in
the command input field, the path must be encapsulated in apostrophes, for example:

```
* import "./cg000000.dat3"
```

If the command is selected from the menu bar, or if no name is specified in the command input field, the Import
dialog box is displayed and the path and file name for a grid file can be be entered. A dialogue confirms that
the current grid should be overwritten by the grid from the specified file.

3.5.18 Saving the grid
The Save and Save as commands in the File menu save the grid, together with the rest of the job data.
Porosities must be recalculated following any change to the grid.

3.5.19 Grid-related utilities
FLACS-CFD is delivered with a command line tool, gm for creating and manipulating the grid. This tool can be
used to quickly edit or obtain information about the grid.

3.6 Porosities menu
The Porosities menu contains commands for calculating and verifying porosities.

- **Calculate** starts the porosity calculation program, FGC (Flacs Geometry Calculator).
- **Verify** starts Flowvis in porosity verification mode.

3.6.1 Calculate
The Calculate command starts the porosity calculation program FGC.
Settings for FGC can be selected in the Preferences dialog in CASD.

3.6.2 Verify
The Verify command lets you view the calculated porosities. Flowvis is started for porosity verification. A
2D Cut Plane plot for the appropriate job is automatically created with volume and area porosities shown.
The Plot Domain dialog box pops up. This dialog box lets you select other planes as wished.
By clicking inside a control volume, you can verify the porosity values for that volume.
3.7 Scenario menu

The purpose of this chapter is to outline how to edit the scenario sections. A short description of each section and the impact on the FLACS-CFD simulations will be given. The scenario file (cs-file) is an ASCII file and it is easy to edit manually as well as using CASD.

The scenario input is done in a scenario panel where sections can be expanded and collapsed. When a section has been expanded, the items in the section are displayed as a list. The method for editing a scenario section depends on the type of section.

Note:

Some advanced options and parameters are hidden by default. These can be shown by right clicking with the mouse and checking Show advanced.

Sections such as INITIAL_CONDITIONS and IGNITION contain a list of parameters, each with one or more values. A parameter is selected for editing by double clicking on it or hitting the enter key. The scenario input can be done using the keyboard only with a combination of the arrow keys and the enter key.

Sections such as SINGLE_FIELD_SCALAR_TIME_OUTPUT and SINGLE_FIELD_3D_OUTPUT contain a list of items, which can be selected.

For some sections, each item has a subsection. For SINGLE_FIELD_SCALAR_TIME_OUTPUT you must select from a list of monitor points for each variable selected. An item is selected by clicking on it. To select several items using the mouse apply CTRL or SHIFT keys. If a selected option shall be deselected without selecting an alternative option, deselect by clicking while pressing the CTRL key.

Importing the scenario from another job

Use the Import command in the File menu menu to import the scenario from another job. You will be asked to verify that the current scenario is overwritten by the scenario from the specified file.

Saving the scenario

The Save and Save as commands in the File menu menu saves the scenario (together with the rest of the job data).

3.7.1 Scenario menu undo/redo

The scenario menu supports undoing and redoing most actions. As a general rule all actions except the Automatic grid refinement plugin support undo/redo.

Undoing is achieved by having focus on the scenario menu and using the CTRL+Z shortcut. Redoing is triggered by using the CTRL+Y shortcut.

There are also two buttons in the toolbar that can be used to trigger undo/redo in the scenario menu. See the screenshot below where these buttons are highlighted.
3.7.2 Scenario settings

3.7.2.1 Simulation type

The following simulation types are available in CASD:

- Dispersion and ventilation
- Gas explosion
- Dust explosion
- Fire
- Inert
- Blast
- Pool

Note:

In FLACS-CFD 21.2 the simulation type Gas explosion (DDT) has been removed. This functionality is covered by the Gas explosion simulation type. Opening old scenarios that use the Gas explosion (DDT) simulation type is still supported in CASD and FLACS-CFD.

3.7.2.2 Write all variables

Enabling this option will write all available variables available in the current simulation type to the SINGLE_FIELD_VARIABLES section of the scenario file. Note that this only concerns variable definitions and does not automatically lead to output being written; see the sections regarding monitor points, pressure relief panels, and 3D output for how to obtain data for the variables.
3.7 Scenario menu

3.7.2.3 cs-file compatibility

This option can be changed to produce scenario files that are compatible with different versions of FLACS-CFD, and will also impact which simulator and porosity calculator is selected when starting simulations using RunManager and runflacs.

Note:

This option is hidden by default. To make it available, right click in the Scenario settings window and check the Show advanced option.

3.7.2.4 Tags

This field is mainly used by FLACS-Risk to assign key-value pairs to scenarios, to support grouping in the post-processing. The input must be a comma-separated string, such as Geometry=BangBox, Wind=0/5.

3.7.2.5 Frequency

The frequency field is used by FLACS-Risk projects to assign a frequency to each scenario. The frequency is used in post-processing to generate exceedance curves.

3.7.3 Monitor points

Monitor points are user-defined locations in the simulation domain where one or more variables are to be monitored during the simulation. The maximum number of monitor points allowed is currently 8000. Positions for monitor points are given in the unit selected in Options Preferences (normally meter).

For the fire simulation type, monitor points additionally need to be assigned a direction along which the incident radiative flux is calculated. The monitor point direction is specified as a vector. If the direction is not specified then the maximum value will be output.

Output variables to be written for a monitor point are selected by editing the monitor point. Several monitor points can be edited at the same time by marking them and clicking 'Edit'.

FLACS-CFD identifies the eight surrounding Control Volume (CV) centres and writes an interpolated value of the specified variables to the scalar-time output file (nodes on the other side of a wall or with zero porosity will not be used when interpolating). Here the eight surrounding CV centres define the corners of a parallelepiped used during the interpolation (for a uniform grid with the same CV length in all three axis directions this parallelepiped will be a cube). When a monitor point is located in the centre of a Control Volume (CV) and output is given for a scalar variable (e.g. pressure) stored during the simulation at the central nodes of the numerical grid (the CV centres), then the value reported as output for the scalar variable is the same as the value at the CV centre itself (except for a possible very small difference due to numerical rounding errors).

Attention:

Monitor points should not be placed in the same grid cell as a solid wall unless the wall is thicker than a single grid cell. Usually it is best to enter monitor points according to the grid, not according to the geometry. During the porosity calculations, walls are adjusted to the nearest control volume face (grid line), which can result in a monitor point that was positioned on one side of a wall now being on the other side of the wall! So, if monitor points are placed on opposite sides of a solid wall, then it is possible that they will be considered in FLACS-CFD to be on the same side of the wall (unless the wall is thicker than a single grid cell). Also ensure that no monitor points are inside fully blocked control volumes. For FLACS-Blast simulations, monitor points should not be in cells with porosity \( \leq 0.5 \), see the blast section in Modelling and application limitations.
Figure 3.18: Specification of monitor points

Figure 3.19: How to position the monitor points

See also:

Monitor points can be added quickly using the Arrange items functionality.

How to obtain output for flow variables at monitor points is described in the section on Single field scalar time output below.

3.7.3.1 Naming monitor points

Monitor points are given the generic name MP # upon creation. The # is replaced by sequential numbering ("1", "2", "3", etc.), converting the generic names into "MP 1", "MP 2", "MP 3", etc. The sequential
numbering is kept consecutive, so that if monitor points in the middle of the list is deleted, monitor points
with higher numbers will be renamed, i.e. assigned new numbers to close the gaps in the numbering. Monitor
points can be given custom names, and such names will not be affected by the automatic numbering, unless
they include a #.

- It is possible to give multiple monitor points the same custom name, but this should be avoided as it
  may lead to confusion later when plotting results.
- Using a custom name that match a generic name (e.g. "MP 5") is also allowed, but may lead to
  multiple monitor points with identical names.
- The monitor points are stored internally in an ordered list, and new monitor points are added to the
  end of this list. The order (index) of the monitor points in this list is fixed, and is not changed when
  renaming or deleting monitor points.
- The sequential numbering for monitor points with names containing a "#" is computed by finding
  the next unused consecutive number when iterating through the internal list. This means that when
  renaming a monitor point from e.g. "MP #" to "INSTRUMENT #", the new name will contain the
  same number as before, but the visual position of the monitor point in the list may change due to
  sorting.
- The visual order of monitor points in the scenario menu is controlled by the chosen sorting criteria,
  independent of the internal list position (index).

3.7.3.2 Grouping of monitor points

Groups can be defined to organise monitor points. All grouping functionality is available in the context
menu, which can be accessed by right-clicking in the monitor points section of the scenario menu. The list
of monitor points in a group can be collapsed by double-clicking on the group name; another double-click
will expand the list again. When marked, one or several monitor points can be dragged into, between or
out of a group. Each monitor point can be assigned to maximally one group. See the picture below for an
example.

3.7.3.3 Labels for monitor points

Aside from organising monitor points in groups, they can also be assigned one or several labels. Labels can
be exploited by filtering the list of monitor points to show only those that have one (or several) specified labels.
See the picture below for an example. The label and filtering functionality is accessed by right-clicking in
the monitor points section of the scenario menu.

![Monitor points screenshot](image)

Figure 3.20: The monitor points in this example have been assigned to two groups representing the decks of
a module. Additionally, labels have been assigned, and a filter is active so that only those monitor points
that have the label *Gas detector* are shown.
Pressure relief panels are commonly used in the process industry as a mitigating device in the case of an explosion. When the pressure forces on the panel exceed a certain limit, the panel yields and the pressure is relieved. In FLACS-CFD, special planar elements inside the simulation volume are used to model the effect of explosion relief panels and yielding walls. They are specified as boxes where one, and only one, of the three side lengths must be zero. In addition, several parameters such as the yield pressures, area porosities and drag factor may be specified.

A panel may be active or passive. An active panel will initially modify the porosity in the region it covers, and again when the pressure difference over the panel exceeds the given limit. If the initial porosity is set to 0 and the final porosity is set to 1 the panel will start as being closed and end up as being open. This is how the behaviour of a real yielding panel is emulated. A passive panel will never modify the porosity, but it may be used to monitor the same data as an active panel. The maximum number of panels allowed is currently 8000. There are 5 different active panel types in standard FLACS-CFD (and additional 2 in flacscfd):

<table>
<thead>
<tr>
<th>Panel type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNSPECIFIED</td>
<td>Panel with linear displacement, full parameter set</td>
</tr>
<tr>
<td>POPOUT</td>
<td>Panel with a linear displacement, limited parameter set</td>
</tr>
<tr>
<td>HINGED</td>
<td>Panel with a rotational displacement, limited parameter set</td>
</tr>
<tr>
<td>PLASTIC</td>
<td>Simulates the presence of plastic sheets (commonly used in experiments)</td>
</tr>
<tr>
<td>OVERLAY</td>
<td>Panel properties are 'blended' with existing equipment</td>
</tr>
<tr>
<td>POPOUT RIGID (flacscfd only)</td>
<td>Rigid panel with a linear displacement, limited parameter set</td>
</tr>
<tr>
<td>HINGED RIGID (flacscfd only)</td>
<td>Rigid panel with a rotational displacement, limited parameter set</td>
</tr>
</tbody>
</table>

The edges of a panel are automatically adjusted to the closest grid line. Specify panels whose dimensions match the grid, either by adjusting the grid or by modifying the panel (see Summary of grid recommendations).

One topic that needs special attention is the presence of solid structures close to the panel area. For example structural beams may constitute quite large blockages that must be accounted for, especially since they also occur at the vent openings of typical offshore modules. In such cases the panel area must be defined using several panels with solid beams in between. Smaller beams, often an integral part of the support frame of the pressure relief panels, may be accounted for by adjusting the final panel porosity for the panel itself.

Apart from the abovementioned types of pressure relief panels, FLACS-CFD supports inactive panels. An inactive panel can be used to monitor variables, e.g. to measure average pressure on surfaces such as decks and walls. See the description of inactive panels.

If several panels are specified, they must not overlap each other anywhere, otherwise the behaviour is undefined; inactive panels are an exception and may overlap.

For panels, the same functionality for defining and using groups and labels is available as for monitor points. The full list of parameters from the scenario file is shown below. The type of the panel determines which of these parameters are relevant.

```
NAME "Panel #"
POSITION 0.1 1.2 30.4
SIZE 7.1 5.5 0
MATERIAL "DefaultMaterial"
PANEL_TYPE "HINGED"
OPENING_PRESSUREDIFFERENCES 100000 0.05
INITIAL_AND_FINAL_POROSITY 0 1
WEIGHT 2.7
DRAG_COEFFICIENT 1
MAXIMUM_TRAVEL_DISTANCE 0
PANEL_SUBSIZES 1.1 4.3
```
3.7 Scenario menu

The parameters for specifying panel properties are described in the following sections.

3.7.4.1 Name
A textual, user-defined name or tag for the panel. The name is saved in the cs-file but not used by the simulator.

3.7.4.2 Insert
Integer identifying the panel.

3.7.4.3 Position
Cartesian coordinates [m] of the corner of the panel (the corner with lowest value of the coordinate in each axis direction).

3.7.4.4 Size
The panel is assumed to be a plane of rectangular shape. The dimension [m] in each of the axis directions is given. One dimension should be zero, this shows how the panel is oriented. If e.g. the dimension in x-direction is zero, the normal of the panel points in the x-direction. The other two dimensions should be positive. The panel edges and the position will be adjusted to match the grid lines perfectly in the FLACS-CFD code, so it is advised that you only specify panels which coincide with the grid lines, in order to avoid any confusion concerning the geometrical representation of the panel.

3.7.4.5 Material
For panels, the materials defined in the CASD database may be assigned, but they will not lead to the use of the associated colour. To change the colour of a panel, the colour has to be given as hue value and must be the only value provided to the MATERIAL property, for example

```
MATERIAL "300"
```

in the cs-file or by clicking Edit in the Pressure relief panels section of the scenario menu in CASD.

3.7.4.6 Panel type
The current section describes the eight different panel types: INACTIVE (the only passive panel type), UNSPECIFIED, POPOUT, HINGED, POPOUT RIGID, HINGED RIGID, PLASTIC, and OVERLAY.

3.7.4.6.1 Inactive panel  A passive panel does not affect the numerical simulation in any respect (neither the geometrical description, including the porosities, nor the flow field), it is only included to monitor variables related to the area which the passive panel occupies (e.g. area-averaged pressure over all the control-volume faces of the panel). Only the parameters INSERT, POSITION, SIZE, and MATERIAL are relevant when the PANEL_TYPE is INACTIVE (in CASD). In the cs-file, a panel is marked as inactive by removing the PANEL_TYPE line in the panel definition. INACTIVE panels are available under Single field scalar time output in the Scenario menu.

3.7.4.6.2 Unspecified, Popout and Hinged panel  An active panel (all panel types except INACTIVE) will generally affect both the porosities representing the geometry and the flow field. For the active panels both an initial and a final area porosity value is given (open area divided by total area). The initial porosity corresponds to the state of the panel before it has started to yield due to external pressure forces. The final porosity corresponds to the state of the panel after it has yielded completely due to pressure forces. Each control-volume face covered by the panel is treated separately. Thus it is possible, depending on the scenario, that one part of the panel is open, while at the same time other parts are closed.
The way of using the initial and final porosities in the FLACS-CFD code depends on the type of panel considered. In the case of the panel types UNSPECIFIED, POPOUT, and HINGED, for each control-volume face that is covered by the panel, the porosity representing the panel is also the porosity of the control-volume face, regardless of any other equipment or structure included in the geometry.

3.7.4.6.3 Popout rigid and Hinged rigid panel POPOUT RIGID and HINGED RIGID panels can be considered to be the more accurate models of POPOUT and HINGED panels (note, validation of the rigid panel type is still ongoing), and is different in the way the area porosity of each control-volume face initially covered by the panel is evaluated. Panel's average area porosity and traveled distance (or opening angle), is updated based on forces and torques acting on the total panel area as it is assumed to be rigid. In case of POPOUT RIGID panels, area porosities of individual control-volume faces are equal to the average panel area porosity, whereas in case of HINGED RIGID panels they depend on face's relative distance to hinges. Hinges position of HINGED RIGID panels must be specified in HINGES. Additionally, for this panel type the user may specify side outflow planes - SIDE OUTFLOW.

The opening angle or the traveled distance of the panel can be visualized in Flowvis or accessed through Python API when PTRA variable is selected in the SINGLE_FIELD_SCALAR_TIME_OUTPUT. The value of PTRA is signed and is positive for translations in the positive axis direction and rotations in the positive direction about the hinge axis.

3.7.4.6.4 Overlay and Plastic panel A panel of type OVERLAY is superimposed on other equipment. For an OVERLAY panel, the value of the area porosity determined by the panel is multiplied by the value of the area porosity determined by other geometrical objects (in the absence of any panel) giving the effective area porosity, for each control-volume face covered by the panel. Let us consider an example where the final porosity of the panel is 1 (fully open). After the panel has yielded completely for all the control-volume faces covered by the panel, the area porosities have the same values that they would have if no panel was included in the geometry. So in this case other geometrical objects than the panel itself can not "blow out" together with the yielding panel (cf. the comments above).

The panel type PLASTIC emulates a plastic sheet (commonly used in experiments). For a PLASTIC panel, in case of initial area-porosity, for each control-volume face which is covered by the panel, the porosity representing the panel is also the porosity of the control-volume face, regardless of any other equipment or structure included in the geometry (as in the case of the panel types UNSPECIFIED, POPOUT, and HINGED). Usually the initial area-porosity of a PLASTIC panel is zero (completely blocked). If the PLASTIC panel has yielded completely, the area porosities have the same values that they would have if no panel was included in the geometry. So in this case other geometrical objects than the panel itself can not "blow out" together with the yielding panel (cf. the comments above).

Panels of the types PLASTIC or OVERLAY are assumed to be light panels, and inertial forces are neglected by the FLACS-CFD code during the dynamical process when the panel yields due to pressure forces. In the case of panels of types UNSPECIFIED, POPOUT, or HINGED, inertial forces are in general taken into account.

3.7.4.6.5 The +IMP panel types The standard panel types will yield at a given negative or positive pressure. In some situations it can be necessary for panels to also take into account the pressure impulse. To activate this, add the suffix +IMP to the end of the panel type string, e.g

```
PANEL_TYPE POPOUT+IMP
OPENING_PRESSURE_DIFFERENCES 0.05 0.005 (bar bar *s)
```

When using this the input to OPENING_PRESSURE_DIFFERENCES changes to

- Parameter 1 is yield pressure for both negative and positive direction [bar]
- Parameter 2 is yield pressure impulse both negative and positive direction [bar*s]

The panel will yield if both the pressure and pressure impulse yield values are exceeded. This is valid for panel types HINGED and POPOUT.
3.7 Scenario menu

3.7.4.7 Opening pressure differences

When the net pressure over the panel (pressure on the negative side relative the coordinate axis minus pressure on the positive side, i.e. the net pressure from the fluid acting on the panel) exceeds the given limits, the panel starts to yield. Two opening pressure-differences, in units of [bar], have to be given by the user, the first corresponds to the case when the net pressure acts in negative direction, the second when the net pressure acts in positive direction. If for example the OPENING_PRESSURE_DIFFERENCES are given by -0.1 and 0.2, the panel starts to yield when the net pressure is less than -0.1 bar (negative direction) or greater than 0.2 bar (positive direction). Notably, the FLACS-CFD simulator disregards the sign of the user-defined opening pressure-differences. Hence, if the OPENING_PRESSURE_DIFFERENCES were given instead by 0.1 and 0.2, or by 0.1 and -0.2, the effect on the numerical simulation would be exactly the same. The opening pressure-differences must be given for the active panels (all the panel types accept INACTIVE).

3.7.4.8 Initial and final porosity

The initial area porosity [-] of the panel (open area divided by total area) corresponds to the state of the panel before it has started to yield due to external pressure forces. Similarly the final porosity corresponds to the state of the panel when it has yielded completely. In the case of a PLASTIC panel, the final area-porosity should be set to 1 in order to correspond to how the FLACS-CFD code works (the final area-porosity is always assumed by FLACS-CFD to be 1, regardless of the given value). How FLACS-CFD uses the initial and final porosities depends on the type of panel considered, see above. The initial and final porosity must be given for the active panels (all the panel types accept INACTIVE). The value of the porosity range from 0 (totally blocked) to 1 (fully open). It is expected that the initial porosity is less than the final porosity (the net pressure opens up the panel). If this is not the case, the FLACS-CFD code gives a warning.

After the panel has started to yield, the effective area porosity will gradually change as function of time until the panel has yielded completely. How long time it takes before the panel has yielded completely depend on the pressure forces acting on the panel, the weight of the panel, and the type of panel (for example will the dynamics when a HINGED panel opens, be somewhat different from the dynamics when a POPOUT panel opens). The panel average porosity PPOR may be monitored as function of time giving some information about the yielding process, confer the description of PPOR.

3.7.4.9 Weight

The parameter WEIGHT is relevant for the panel types POPOUT, POPOUT RIGID, HINGED, and HINGED RIGID. This parameter is defined as the mass per unit area of the part of the panel that is moved when the net pressure acting on the panel exceeds the given opening pressure-difference; (mass of moved area of panel)/(moved area of panel) [kg/m²]. The denominator (moved area of panel) is not the total area, but only the area of the moved part of the panel (in general this is less than the total area). The dynamics when the panel yields and part of the panel is accelerated and move away from its initial position, is modelled in the FLACS-CFD code. When specifying a panel type that includes the WEIGHT parameter, the inertial forces must be there when the panel yields. This means that value of WEIGHT should be set to a positive number greater than zero, or the FLACS-CFD simulator will terminate with an error message in the log file. It is only the part of the panel that is accelerated and moves away from its initial position that should contribute to the mass of the moved area of the panel. The mass of a rigid frame that does not yield, should not be included.

Let us consider an example with a POPOUT pressure relief panel: Uniform steel plates with mass per unit area 10 kg/m² are mounted on a strong frame. When the pressure difference exceeds a certain level, the steel plates pop out and move away from the frame. The final porosity corresponds to the state when the panel has yielded completely (the steel plates have moved sufficiently away). Depending on the structure of the strong frame (assumed not yielding), the final porosity could e.g. be 0.9. And in this example the parameter WEIGHT is given as input 10 kg/m² (the value 10 kg/m² is within the typical range of values for WEIGHT for pressure relief panels used at offshore installations).
3.7.4.10 Drag coefficient

A dimensionless drag-coefficient [-] has to be given for the panel type UNSPECIFIED (for the other panel types it is not relevant to specify this parameter). The drag coefficient is used when the drag force from the panel on the fluid is modelled (both before, during, and after the panel has yielded). The value of the DRAG_COEFFICIENT is zero or positive. A typical value is 2.0 (the value 2.0 is set by the FLACS-CFD code as a fixed preset value for the panel types POPOUT and HINGED).

3.7.4.11 Maximum travel distance

The maximum travel-distance [m] has to be given for the panel type UNSPECIFIED (not relevant for the other panel types). The use of the maximum travel-distance is based on a rough approximation. The maximum travel-distance is used to model the effective area-porosity at the initial position of the panel after the panel yields. A typical value of the MAXIMUM_TRAVEL_DISTANCE is in the order of 1m (for the panel types POPOUT and HINGED there is an in-built model in the FLACS-CFD code for the effective area-porosity that does not need an input value of the MAXIMUM_TRAVEL_DISTANCE).

You do not need to explicitly specify the maximum travel distance [m] for a POPOUT or HINGED panel. For a POPOUT panel, the maximum travel distance $d_{\text{max}}$ [m] (i.e. the panel displacement after yielding, where the panel frame is fully open, corresponding to the final porosity) is calculated from the sub-panel width $w$ [m] and the sub-panel height $h$ [m] according to the formula:

$$d_{\text{max}} = \frac{w \cdot h}{w + h}.$$  \hspace{1cm} (3.1)

Here the width $w$ is the first panel subsize, and the height $h$ is the second panel subsize (see below). For example, if the panel subizes are specified as ‘PANEL_SUBSIZES 2.0 1.0’, then the width is 2.0 metres and the height is 1.0 metre.

A POPOUT panel covers a certain number of Control Volume (CV) faces for the given numerical grid. As a simplification (or model limitation), the panel displacement at each CV face is assumed to be independent of the other CV faces. The force acting at each CV face is $F = A \cdot dp$ [N], where $A$ [m$^2$] is the effective area and $dp$ [Pa] is the local pressure difference (drag forces are also accounted for). The velocity and displacement at each CV face is updated for every time step. The acceleration $a$ [m/s$^2$] is calculated as $a = F/m$ where $m$ [kg] is the mass of the segment of the yielding panel that covers the CV face. Integrating the acceleration $a$ with respect to time gives the velocity $v$, and integrating the velocity $v$ over time gives the panel displacement $d$ as function of time after yielding. The area porosity at each CV face covered by the POPOUT panel will change linearly from the specified initial porosity to the specified final porosity as the panel moves from zero to the maximum travel distance.

3.7.4.12 Panel sub-sizes

The parameter PANEL_SUBSIZES has to be defined for the panel types POPOUT or HINGED (not relevant for the other panel types). These panel types are assumed to consist of sub-panels mounted on a frame. The width [m] and the height [m] of the sub-panels need to be given (it is assumed in the numerical model that all the sub-panels are of uniform size). Both the width and the height should be larger than zero. In the case of the panel type HINGED it is assumed that each sub-panel turns on a hinge when it yields. The width of the sub-panel is defined as the dimension in the direction normal to the axis of the hinge. It is important (since it affects the model in FLACS-CFD) to specify the width of a HINGED panel first and then the height, in the parameter PANEL_SUBSIZES (for a POPOUT panel it is of no importance which dimension that is defined as the width).

See also:

Panels can be added quickly using the Arrange items functionality.

How to obtain output of flow variables on panels is described in the section on Single field scalar time output below.
3.7 Scenario menu

3.7.4.13 Hinges (flacscfd only)

The location of the hinges can be specified only for **HINGED RIGID** panels. Parameter **HINGES** takes one of the values denoting hinges axis position with respect to the panel. For instance for the **XY** panels the following values are allowed: \(+X, -X, +Y, -Y\).

3.7.4.14 Side outflow (flacscfd only)

Side outflow can be specified for **HINGED RIGID** panels. By default (None) the model uses the most conservative approach, where only the projected panel opening area is taken into account in average panel porosity calculations. The user may change this behaviour and select also Frontal, Lateral or both areas. Taking into account both areas is the most accurate modelling approach for panels, assuming their corresponding areas are free and do not interfere with neighbouring panels. When in doubt, it is recommended to use the most conservative approach (with the smallest opening area).

3.7.4.15 Using panels in fire simulations

All panel types, including pressure relief panels, are supported by the FLACS-Fire solver. The DTM radiation calculations will capture any changes to panel porosity during the simulations (e.g. due to panels failing or when using an event file).

3.7.5 Single field scalar time output

Rather than having its own entry in the scenario menu, the output of values at monitor points and panels is selected directly when defining the **monitor point** or **panel** in their respective scenario menu sections. One or several variables can be selected and deselected by left-clicking or dragging, respectively. Typically the same set of variables should be measured at all monitoring locations. This can be done quickly by selecting all monitor points/panels, right-clicking and selecting 'Edit' on the context menu. The variables can then be selected for all selected monitor points/panels.

An example of how the single field scalar time output selection translates to the corresponding section in the scenario file is shown below. Here **NP** identifies the variable **P** (pressure) whereas **NPP** selects the variable **PP** (panel pressure), which gives the average pressure across a the surface of a panel:

```
SINGLE_FIELD_SCALAR_TIME_OUTPUT
NP 1 2 3 4 5
NPP 1 2 3
```

This shows that pressure is reported for monitor points 1-5 and panel pressure is reported for panels 1-3. The most commonly used monitor point variables for explosion simulations are pressure (**P**), dynamic pressure (**DRAG**), panel pressure (**FP**), pressure impulse (**P_IMP**) and sometimes flow velocity (**UVW**). For dispersion calculations volume gas concentration (**FMOLE**) and flow velocity (**UVW**) are often used.

The monitor point results for job 0101000 are written to the r1010100.dat3 file which is read by Flowvis. If ASCII data is required, the rfile-utility program can be used (see the section on FLACS-CFD utilities).

The first section in the scenario file defines the names and identifiers for all variables which may be selected for output from FLACS-CFD. To select alternative units for certain variables (e.g. psi or kPa for pressure or K for temperature) the scenario file should be edited manually. The variable pressure is described in the scenario file as follows:

```
NP "P " 1 "(barg) " N "Pressure"
```

The output can be obtained in a different unit (psig) can be obtained by editing this as follows:

```
NP "P " 1 "(psig) " N "Pressure"
```

Similar changes can be made for other variables and other units. Notably the unit of time must always be seconds in the scenario file. After running a simulation, it is possible to change the unit of time as well as other units in Flowvis.

A complete list of all variables available can be found in section Output variables in FLACS-CFD.
3.7.6 Single field 3D output

This is an output facility in FLACS-CFD which makes it possible to generate plots of the spatial distribution of the variables (e.g., cut plane plots and volume plots) at different moments in time. Before running a simulation, the user needs to specify the list of desired variables for **SINGLE_FIELD_3D_OUTPUT**, an example from scenario file is shown below (here **P**, **PROD** and **VVEC** are selected in CASD):

```
SINGLE_FIELD_3D_OUTPUT
NP
NPROD
NVVEC
NU
NV
NW
```

When velocity vectors are selected for output (**VVEC**), the velocity components **U**, **V** and **W** will automatically be selected. These must not be deselected while **VVEC** remains selected (this might, for example, happen when editing the scenario file manually). Failure to keep the result file and the scenario file consistent will lead to strange results shown in Flowvis.

This type of output may give very large files (**r3-files**). If you want to save disk space, the number of output variables and the number of time instants for output must be limited. The **r3-files** are binary, if ASCII data is wanted the utility program **r3file** can be applied.

The most commonly used variables for explosion modelling are pressure (**P**), flame (**PROD**), sometimes gas volume concentration (**FMOLE**), dynamic pressure (**DRAG**), maximum pressure (**P_MAX**) and velocity vectors (**VVEC**). For dispersion **FMOLE** and **VVEC** will be the most common variables to report.

In certain situations the variable **P_MAX** may not be written to even if specified (zero results everywhere). This may happen if the simulation job requires more RAM than allowed (e.g., due to self-defined limits in Linux).

To specify output times **DTPLOT** and **NPLOT** will normally be used (see Simulation and output control section), and sometimes also cc-files (see Runtime simulation control file section). To create animations it is normally recommended to have plots at 100-200 different moments in time. When creating results files to be used for animations a combination of **DTPLOT** and **NPLOT** is usually recommended for explosions, for dispersion only **DTPLOT** should be used (i.e., keep **NPLOT=-1**).

Units of output variables can be changed using the technique described in the previous section. A complete list of all variables available can be found in section Output variables in FLACS-CFD.

3.7.7 Setup

This section of the scenario menu has to be enabled manually and allows to specify the advanced settings that can be provided for a simulation with the help of a setup file. The setup file written by CASD will have the name `cs<jobno>.SETUP` and will be read automatically by the Flacs simulator unless a setup file (with a different name) is specified explicitly on the command line call to Flacs.

3.7.8 Simulation and output control

This section describes parameters for general simulation and output control. The settings entered into the scenario file by default for gas explosion simulations are listed below:

- **Maximum time** -1
- **Last timestep** -1
- **CFLC** 5
- **CFLV** 0.5
- **MODD** 1
- **NPLOT** -1
- **DTPLOT** -1

When using **flacsref**, the following parameter is also available:
3.7 Scenario menu

Solver mode "TRANSIENT"
Enable DDT detection "AUTO"
Force DDT at time -1

When Solver mode is set to STEADY, the following parameters are also available:

UNDERRELAX_P 0.05
UNDERRELAX_UVW 0.8
CHECK_CONVERGENCE 1
CONVERGENCE_TOLERANCE_P 0.01
CONVERGENCE_TOLERANCE_FUEL 0.001
CONVERGENCE_TOLERANCE_FLAMMASS 0.001
CONVERGENCE_TOLERANCE_FUELMASS 0.001
CONVERGENCE_TOLERANCE_FUELRATE 0.001
CONVERGENCE_TOLERANCE_MASSRATE 0
CONVERGENCE_TOLERANCE_VELMAX 0.0001

If Show advanced is checked, the following parameters are also available:

Start time -1
Minimum time -1
Load number -1
Load job name ""
Timestep code ""
Special control keys ""
INCOMPRESSIBLE 0
WALLF 1
HEAT_SWITCH 0

For dispersion simulations, higher values (20 and 2) are recommended for the CFL numbers. NPLOT should be -1 and a finite value of DTPLOT should be given.

A detailed description of each parameter is given below.

3.7.8.1 Start time (TSTART)

Note:

Only available if Show advanced is checked.

The value entered for Start time is saved in the scenario file as TSTART. This variable makes it possible to specify a start time for simulation (-1 means not applied => default is zero or time of dump-file). If a dump-file exists, one can still adjust Start time, but the previous history of the simulation cannot then be stored using the KEEP_OUTPUT option in a setup file.

3.7.8.2 Minimum time (TMIN)

Note:

Only available if Show advanced is checked.

The value entered for Minimum time is saved in the scenario file as TMIN. This variable makes it possible to define a minimum time for the simulation. When TMIN is set to a positive number, the automatic stop criteria will not activate before the physical time has reached the value of TMIN. This can be useful for blast simulations.

If TMIN is set to -1, then TMIN will not affect the activation of the automatic stop criteria. However, whether TMIN is defined at all, or is set to -1, will in general affect whether FLACS-CFD runs until the value of TMAX (Maximum time) is reached or stops earlier, at a physical time less than TMAX.

Note:

The automatic stop criteria are disabled in the FLACS-Fire solver, so setting TMIN has no effect for these simulations.
3.7.8.3 Maximum time (TMAX)

The value entered for **Maximum time** is saved in the scenario file as **TMAX**. This is the maximum time interval (seconds) that the simulation will last. The default value of -1 is generally appropriate for explosions simulations. Setting **TMAX=-1** means that no maximum time is specified, and the automatic stop criteria will be applied. The automatic stop criteria work well for most explosion calculations. A simulation will be stopped automatically when ignition has happened and one of the following criteria are met:

- at least 90 % of the fuel has been burnt or left the domain, or
- at least 50 % of the fuel has been burnt or left the domain and the average pressure becomes negative

As soon as one of these criteria is fulfilled, the remaining time to be simulated is taken to be 20 % of the time that has passed since the ignition in the simulation. After this time the simulation will stop automatically.

There are no automatic stop criteria for dispersion and fire simulations, and so these will run indefinitely (until manually killed by the user) if **Maximum time** is set to -1 for these scenarios.

Automatic stop criteria may not be useful for all explosion scenarios. If far-field blast pressures are of interest, the automatic stop criteria should not be used as they may stop the simulation before blast waves have hit their target.

In a gas dispersion or fire simulation, an appropriate **TMAX** will typically be from a few seconds to a few minutes. Simulation results are not affected when varying this parameter.

When **TMIN** (Minimum time) is not specified in the scenario file, and **TMAX** is set to a positive value less than 10 000, then FLACS-CFD will not stop due to the automatic stop criteria, but will run (whenever possible) until the physical time reaches the value of **TMAX**. However, when the **TMIN** is specified in the scenario file, (even when **TMIN=-1**), then FLACS-CFD may stop before the physical time reaches **TMAX** (provided that **TMAX** is set to a positive value larger than **TMIN** but less than 10 000). If **TMIN** is set to a positive value larger than the time determined by the automatic stop criteria, then FLACS-CFD will run (whenever possible) until the time reaches the value **TMIN** and then stop the simulation. If **TMIN** is set to a value less than the time of the automatic stop criteria, then FLACS-CFD will run (whenever possible) until the automatic stop criteria are met.

Some examples of different **Minimum time** and **Maximum time** settings follow:

### 3.7.8.3.1 Example 1: Typical explosion simulation

Automatic stop criteria are used if the cs-file has these settings:

```
TMIN -1
TMAX -1
```

### 3.7.8.3.2 Example 2: Explosion simulation with points of interest in the medium/far field

In such cases, use **STEP="KEEP_LOW"** to keep the time step small after the explosion and to prevent smearing of the pressure waves (see **Timestep code**). Example cs-file setting are:

```
TMIN 3
TMAX -1
```

These settings will stop the simulation at 3 s unless more than 10 % of the fuel is still present. The appropriate **TMIN** (Minimum time) depends on the distance to the point of interest and the time of the explosion peak pressure.
3.7 Scenario menu

3.7.8.3.3 Example 3: Dispersion or Fire simulation

By prescribing

- \( T_{\text{MIN}} = -1 \)
- \( T_{\text{MAX}} = 100 \)

the simulation will stop at 100 s.

3.7.8.4 Last timestep (LAST)

Note:

Only available if \textit{Show advanced} is checked.

The value entered for \textit{Last timestep} is saved in the scenario file as \textit{LAST}. This stop criterion is based on the maximum number of iterations allowed for the simulation. The default value set by CASD is \(-1\), which means that there is no limitation to the number of iterations. This parameter is particularly useful when the steady state solver is used. The dump snapshot is written automatically when the \textit{LAST} iteration is reached and the simulator exits normally.

3.7.8.5 Load number (LOAD)

Note:

Only available if \textit{Show advanced} is checked.

The number entered for \textit{Load number} is saved in the scenario file as \textit{LOAD}. This setting makes it possible to load a dump snapshot from a file (either from the CGNS file or the rd-file). It is equivalent to legacy option \textit{NLOAD} in the cc-file. The default value set by CASD is \(-1\), which means that no snapshot will be loaded. Otherwise, the dump snapshot denoted by \textit{LOAD} will be loaded. If \textit{LOAD} is set to 0, then the simulator will load the last available dump snapshot.

3.7.8.5.1 Example 1 - load snapshot #2 into simulation 000000

Load number setting in the cs-file:

- LOAD 2

What happens:

1. Simulator first tries to read "rd000000.n002", which is an obsolete Flacs2 dump file containing snapshot #2.
2. If no rd-file is found and the simulator is Flacs2, then the simulator will exit with an error message.
3. If no rd-file is found and the simulator is Flacs3, then the simulator tries to read dump snapshot #2 from the CGNS file "000000.cgns".
4. If any of the files are read, then the simulator restores the exact conditions from the time when the file was written.
5. Simulation output is appended to the existing result files.
6. An error message is given if neither "rd000000.n002" nor the CGNS file exist.
3.7.8.5.2 Example 2 - load the last available dump snapshot from the CGNS file into simulation 000000

Load number setting in the cs-file:

LOAD 0

What happens:

1. The last available dump snapshot from the CGNS file "000000.cgns" is read at simulation startup, restoring the exact conditions from the time when the file was written.

2. Simulation output is appended to the existing result files.

3. An error message is given if the CGNS file does not exist.

3.7.8.6 Load job name (JLOAD)

Note:

Only available if Show advanced is checked in flacscfd.

The string entered for Load job name is saved in the scenario file as JLOAD. This makes it possible to change the name of the CGNS file that the dump snapshot is loaded from. When JLOAD is specified in the cs-file, the simulator appends ".cgns" to the end of the provided string, and loads the dump snapshot from the cgns file that corresponds to the resulting filename. When JLOAD is not specified in the cs-file, the current job number is used.

Note:

When grids defined in the grid file differ from those in the CGNS file that contains the dump snapshots, the name of the CGNS file must be set to the string entered for JLOAD, and this must differ from the current job name.

If JLOAD is set, then the history of the simulation cannot be kept using KEEP_OUTPUT in a setup file.

3.7.8.6.1 Example 1 - load from specified CGNS file into simulation 000000 and scale if necessary

LOAD and JLOAD settings in the cs-file:

LOAD 2
JLOAD "dispersion_results"

What happens:

1. Dump snapshot #2 from the file "dispersion_results.cgns" is read at simulation startup, restoring the exact conditions from the time that the file was written.

2. Results are projected onto the new grid if necessary.

3. An error message is given if the file "dispersion_results.cgns" does not exist.

4. A new CGNS results file, "000000.cgns", is created for the job and "dispersion_results.cgns" is not modified.
3.7 Scenario menu

3.7.8.7 CFLC

This is a Courant-Friedrich-Levy number based on sound velocity. The value of CFLC connects the simulation time step length to the control volume size through signal propagation velocity (in this case the velocity of sound), in the following way:

Each time step length is chosen so that sound waves may propagate only a limited distance, which is the average control volume length multiplied by the value of CFLC. The default value set by CASD is CFLC=5 for explosion simulations. The time step limit imposed by this criterion is normally dominant in the early phase of an explosion, when flow velocities and combustion rate are still low (see also CFLV).

Simulation results may change with this parameter, and it is therefore not recommended to change its value for explosion simulations (since this would nullify Validation). If convergence problems occur (a rare occurrence), then CFLC may be reduced by a factor of 2. However, other problems should be ruled out first. Large changes to the CFL number (e.g., an order of magnitude) are never recommended for normal simulations.

For dispersion and ventilation simulations, the default value of CFLC=10 is recommended. This can be increased by the grid refinement factor (if applicable). If the grid is refined near the leak by a factor of 5, then using a CFLC number of 10*5=50 may be considered (a lower value should be used in case of stability problems).

For fire simulations, the default value of CFLC=20 is recommended, however if stability issues are encountered then this can be halved to make the calculations more stable.

For far-field blast simulations, this should be combined with STEP="KEEP_LOW" to keep the time step short, even after the explosion is outside the "core" area, to reduce smearing of the pressure waves (more information is given Timestep code).

3.7.8.8 CFLV

This is a Courant-Friedrich-Levy number based on fluid flow velocity. The value of CFLV connects simulation time step length to control volume dimension through signal propagation velocity (in this case the fluid flow velocity), in the following way:

Each time step length is chosen so that the fluid may propagate only a limited distance, which is the average control volume size multiplied by the Courant number. The default value set by CASD is CFLV=0.5 for explosion simulations. The time step limit imposed by this criterion is normally dominant in the later phase of an explosion, when flow velocities and combustion rate are high (see also CFLC).

Simulation results may change with this parameter, and it is therefore not recommended to change its value for explosion simulations (since this would nullify Validation). In rare cases where convergence problems occur, then CFLV may be reduced by a factor of 2. However, other problems should be ruled out first. Large changes to the CFL number (e.g., an order of magnitude) are never recommended for normal simulations.

For dispersion and ventilation simulations, the default value of CFLV=1 is recommended, and for fire simulations, the default value of CFLV=2 is recommended. A lower value should be used in case of stability problems.

Some additional considerations are required when setting CFLV for the incompressible solver, these are detailed in Time step for the incompressible solver.
3.7.8.9 Solver mode (MODE)

**Note:**

Only available in flacscfd.

The value entered for *Solver mode* is saved in the scenario file as *MODE*. Solver mode is set to TRANSIENT by default which is the recommended setting for most simulations, especially those involving transient phenomena. If the solver mode is set to STEADY, the steady-state solver is activated. Steady-state solver switches off the CFL stability criterion in the time step determination and ignores the CFLC and CFLV settings. The steady-state solver ignores the TMAX (Maximum Time) stop criterion by default, and uses the convergence stop criteria if CHECK_CONVERGENCE (Check convergence) is enabled. Additionally, the maximum number of iterations can be set using the LAST (Last timestep) stop criterion. FLACS-CFD writes the dump snapshot automatically on either convergence or reaching the Last timestep stop criterion.

**Note:**

Using the steady state solver, a steady state solution can be obtained in a fraction of time required for simulations in transient mode. However, in some cases the solution may diverge. This was observed in simulations involving high mass flow diffuse leaks.

3.7.8.10 Enable DDT detection (DDT)

**Note:**

Only available in flacscfd.

The value entered for *Enable DDT detection* is saved in the scenario file as *DDT*. The default value set by CASD is AUTO, which means that detonation will be triggered automatically for hydrogen explosions if DPDX (DPDX) is higher than 1. If *Enable DDT detection* is set to Off detonation will not be triggered. The time at which DDT is triggered can be modified using DDT_TIME (Force DDT at time).

The detonation solver is an engineering approach to simulate detonations. When detonation is activated the maximum speed is allowed to exceed sonic speed and FLACS-CFD can represent supersonic flow. The time step is reduced by a factor of 5 and the burning velocity is set to a constant value.

In order to simulate detonation for other fuels than hydrogen, *Enable DDT detection* has to be set to On.

3.7.8.11 Force DDT at time (DDT_TIME)

**Note:**

Only available in flacscfd.

The value entered for *Force DDT at time* is saved in the scenario file as DDT_TIME. The default value set by CASD is -1, which means that the detonation solver will be triggered for hydrogen explosions when DPDX > 1. If a DDT_TIME is specified, the detonation solver will be triggered independently of the DPDX value and the fuel used.

Some examples of different *Enable DDT detection* and *Force DDT at time* settings follow:

3.7.8.11.1 Example 1: Hydrogen explosion simulation with automatic trigger of detonation

Automatic trigger of detonation when DPDX exceeds 1 if the cs-file has these settings:

```
DDT "AUTO"
DDT_TIME -1
```

Example 2: Hydrogen explosion simulation with specific trigger of detonation

Automatic trigger of detonation when time exceeds 0.02 seconds if the cs-file has these settings:

```
DDT "AUTO"
DDT_TIME 0.02
```
3.7 Scenario menu

3.7.8.11.2 Example 3: Non-hydrogen explosion simulation with automatic trigger of detonation

Automatic trigger of detonation when \( \text{DPDX} \) exceeds 1 if the cs-file has these settings:

\[
\text{DDT} \ "\text{On}\" \\
\text{DDT\_TIME} \ -1
\]

Note: Automatic trigger of detonation for non-hydrogen explosion scenarios is not recommended, since the \( \text{DPDX} \) threshold would be highly conservative for some fuels (for instance, hydrocarbons).

3.7.8.11.3 Example 4: Non-hydrogen explosion simulation with specific trigger of detonation

Automatic trigger of detonation when time exceeds 0.02 seconds if the cs-file has these settings:

\[
\text{DDT} \ "\text{On}\" \\
\text{DDT\_TIME} \ 0.02
\]

3.7.8.12 Underrelaxation \( P \) and \( UVW \) (UNDERRELAX\_\*)

Note:

Only available when Solver mode (MODE) is set to STEADY in flacscfd.

The values set for Underrelaxation \( P \) and \( UVW \) are saved in the scenario file as UNDERRELAX\_P and UNDERRELAX\_UVW, respectively. These two parameters set numerical underrelaxation factors for continuity and momentum equations. They should be left unchanged unless user experiences stability or convergence problems. The stabilising solution describes how UNDERRELAX\_P and UNDERRELAX\_UVW can be used to stabilize the solution.

3.7.8.13 Check convergence (CHECK\_CONVERGENCE)

Note:

Only available when Solver mode (MODE) is set to STEADY in flacscfd.

Enabling this setting by checking the box in CASD writes CHECK\_CONVERGENCE=1 to the cs-file. This activates a convergence stop criterion. By default, the criterion is enabled when the steady-state solver is used, and disabled when the transient solver is used. If the user sets CHECK\_CONVERGENCE=0 for a simulation using the steady-state solver, then only the LAST (Last timestep) stop criterion is used. The dump snapshot is written automatically on convergence and the simulator exits normally.

3.7.8.14 Convergence tolerance (CONVERGENCE\_TOLERANCE\_\*)

Note:

Only available if CHECK\_CONVERGENCE (Check convergence) is enabled and MODE=STEADY (Solver mode) in flacscfd.

The value set for Convergence tolerance is saved in the scenario file as CONVERGENCE\_TOLERANCE\_\*. Convergence tolerance can be defined for the following convergence criteria:

- \( \text{CONVERGENCE\_TOLERANCE\_P} \) - residual of the pressure correction (continuity) equation;
- \( \text{CONVERGENCE\_TOLERANCE\_FUEL} \) - residual of the fuel equation;
- \( \text{CONVERGENCE\_TOLERANCE\_FLAMMASS} \) - smoothed residual of the flammable mass in the default monitor region normalized by its maximum value;
- \( \text{CONVERGENCE\_TOLERANCE\_FUELMASS} \) - smoothed residual of the fuel mass in the default monitor region normalized by its maximum value;
- **CONVERGENCE_TOLERANCE_FUELRATE** - smoothed residual of the fuel rate in the default monitor region normalized by the leak rate (only checked if any leak found);

- **CONVERGENCE_TOLERANCE_MASSRATE** - smoothed residual of the mass rate in the default monitor region normalized by the leak rate (only checked if any leak found);

- **CONVERGENCE_TOLERANCE_VELMAX** - smoothed residual of the maximum velocity in the default monitor region normalized by its maximum value.

The default values of convergence tolerance are conservative and defined such that in most cases the results should converge to the steady state solution in reasonable time. However, experienced users are encouraged to change them. Setting value of the convergence tolerance to 0 disables checking convergence for the respective criteria.

### 3.7.8.15 Timestep code (STEP)

**Note:**

Only available if **Show advanced** is checked.

The value set for **Timestep code** is saved in the scenario file as **STEP**. It is possible to specify different time stepping options, normally to impose additional restrictions on the time step length. The values accepted by **STEP** are summarised in the following table.

Table 3.22: Options regarding the time stepping, specifiable with the **STEP** keyword in the cs-file. The options having a **=** require a real value of the unit specified in the table.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_MAX=</td>
<td>upper bound for the sound wave velocity to be used in computing the time step</td>
<td>[m/s]</td>
</tr>
<tr>
<td>DT_FIX=</td>
<td>fixed time step to be used, overwrites CFL criteria etc.</td>
<td>[s]</td>
</tr>
<tr>
<td>DT_MIN=</td>
<td>lower bound for the time step</td>
<td>[s]</td>
</tr>
<tr>
<td>KEEP_LOW=</td>
<td>prevent the time step from growing when explosion calms down; recommended</td>
<td></td>
</tr>
<tr>
<td></td>
<td>for calculations of far-field blast propagation</td>
<td></td>
</tr>
<tr>
<td>L_FIX=</td>
<td>use the specified grid-cell size as basis for time step determination and</td>
<td>[m]</td>
</tr>
<tr>
<td></td>
<td>ignore possible local grid refinement; this can be used instead of increasing</td>
<td></td>
</tr>
<tr>
<td></td>
<td>the CFL number as a result of grid refinement</td>
<td></td>
</tr>
<tr>
<td>V_MIN=</td>
<td>lower bound for the flow velocity to be used in computing the time step</td>
<td>[m/s]</td>
</tr>
<tr>
<td></td>
<td>from CFL; prescribing <strong>V_MIN</strong> switches off the CFL criterion in the time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>step determination; for the incompressible solver the default is <strong>V_MIN=1</strong></td>
<td></td>
</tr>
</tbody>
</table>

### 3.7.8.16 Special control keys (KEYS)

**Note:**

Only available if **Show advanced** is checked.

Any values entered as **Special control keys** are saved as **KEYS** in the scenario file. This provides the option for entering **setup file** options directly in CASD, e.g.

```
AMUL=0.6e-5
```

sets new value for the laminar viscosity. Multiple values are accepted using comma separation in the cs-file:

```
KEYS = "PR_BUO=0.9,AMUL=0.6e-5"
```

This string sets a new value for the Prandtl-Smith number in buoyancy term and sets a new value for the laminar viscosity (default AMUL=2.0e-5).

```
KEYS="PS1=01"
```

removes the criterion in FLACS-CFD that the maximum speed cannot exceed sonic speed (imposed due to the porosity concept) and ensures that FLACS-CFD can represent supersonic flow.
This is essential for the proper description of the shock wave that results from a vessel burst with high initial pressure. 

**KEYS="RESET=11"** ensures the consistent conversion of the three variables MIX, H, and TAU (cf. the guidance about the conversion option for rdfile utility), and also resets the turbulence level to quiescent (either the default or specifically set values of KIN0, EPS0). **KEYS="RESET=10"** also requires consistent conversion of the three variables, but turbulence is **not** reset to quiescent. Finally, **KEYS="RESET=01"** resets the turbulence level to quiescent, but no consistent conversion of the three variables MIX, H, and TAU is carried out.

**KEYS="KIN0=1.0E-03"** sets the quiescent level for turbulent kinetic energy per unit mass (here, for example, to the value $10^{-3} \text{[m}^2/\text{s}^2\text{]}$).

**KEYS="EPS0=1.0E-03"** sets the quiescent level for the rate of dissipation of turbulent kinetic energy per unit mass (e.g. to the value $10^{-3} \text{[m}^2/\text{s}^3\text{]}$ in the above specification).

It is possible to change the constants in the $k - \varepsilon$ turbulence model by setting new values for:

- **CMY** Changes $C_\mu$ in the equation for effective viscosity. Default value = 0.09.
- **CEPS1** Changes $C_{1e}$ in the equation for production of dissipation. Default value = 1.44.
- **CEPS2** Changes $C_{2e}$ in the transport equation for dissipation. Default value = 1.92.
- **CEPS3** Changes $C_{3e}$ in the equation for production of dissipation. Default value = 0.8.
- **PR.K** Changes $\sigma_k$ in the transport equation for turbulent kinetic energy. Default value = 1.0.
- **PR.EPS** Changes $\sigma_\varepsilon$ in the transport equation for dissipation. Default value = 1.3.
- **PR.FMIX** Changes $\sigma_{\text{mix}}$ in the transport equations for mixture fraction, fuel and enthalpy. Default value = 0.7.
- **PR.BUO** Changes $\sigma_b$ in the equation for turbulent production due to buoyant forces. Default value = 0.9.

**Warning:**

FLACS-CFD is only validated using its default values. Do not change the constants in the turbulence model and transport equations without good reason and careful consideration.

### 3.7.8.17 MODD

This is a parameter that may be used to determine how often data for scalar-time plots are written to the results file during a simulation: data are namely stored every MODD time steps. CASD default is set to 1. This variable does not influence simulation results, only the amount of data stored.

This is normally not used in explosion simulations, but a value of MODD=10 (or higher) may be used for long dispersion simulations.

### 3.7.8.18 NPLOT

This is a parameter that may be used to determine how often data for field plots are written to file during a simulation: data are namely stored at given fuel levels where NPLOT is the number of fuel levels equally spaced between zero and a maximum. Fuel level is defined as the current total mass of fuel divided by the initial total mass of fuel. This output mechanism is not active in the case of a gas dispersion simulation (leaks are specified). This variable does not influence simulation results, only the amount of data stored.

### 3.7.8.19 DTPlot

This is the time interval (in seconds) for field output. This is useful in gas dispersion simulations and also in gas explosion simulations when frequent output is required. The field output file will become very large if DTPlot is set small. This variable does not influence simulation results, only the amount of data which is stored.
3.7.8.20 Incompressible

Note:

Only available if Show advanced is checked.

This parameter turns on the incompressible solver in Flacs.

3.7.8.21 Wall functions (WALLF)

Note:

Only available if Show advanced is checked.

This setting is saved in the scenario file as WALLF. This is a control switch that specifies the use of wall-functions in FLACS-CFD. Wall-functions are used to resolve the effect of momentum and thermal boundary layers on the momentum and energy equations in near wall regions. In the scenario file the following turns Wall-functions on and off:

- 0 = OFF
- 1 = ON

The CASD default value of WALLF is set to 1. This parameter will influence the simulation results. When WALLF=1, wall functions are implemented, following the theory in (Sand & Bakke, 1989). A slightly modified version of this wall-functions procedure is employed when WALLF=2. However, no validation work for WALLF=2 is available.

The default value of WALLF=1 should always be used.

3.7.8.22 Heat switch (HEAT_SWITCH)

Note:

Only available if Show advanced is checked.

This setting is saved in the scenario file as HEAT_SWITCH. This parameter controls activation of the thermal attributes on objects in Flacs. Except for fire simulations, the default choice is off as large scale explosions are not much influenced by heat loss. Turning on HEAT_SWITCH will let walls and objects have background temperature, and if gas temperature changes, some heat transfer into or out of gas will take place. This is useful for small-scale confined explosions and dispersion with important heat effects. This can be combined with radiation models to account for radiative heat losses. For fire simulations the default choice is on. When deactivated, no heat radiation is calculated.

If HEAT_SWITCH is activated, all solid surfaces will be initialised with the ambient temperature. Further heat objects can be specified at different temperatures using the Heat file.

For both Gas explosion and fire simulations, the radiation model can be activated in the separate Radiation scenario section. When the Radiation model is enabled, the HEAT_SWITCH is activated in the solver (override the cs-file settings).

See also:

Refer to Radiation model for a more detailed description.

3.7.8.23 Split CGNS (SPLIT_CGNS)
3.7 Scenario menu

Note:

Only available if Show advanced is checked.

This setting is saved in the scenario file as SPLIT_CGNS. This parameter controls splitting of the CGNS file. If splitting is disabled, then all data are written to a single <JOB>.cgns. By default, splitting is enabled for all 1D, 3D and Dump output. When splitting is enabled, the main CGNS file <JOB>.cgns contains links to many smaller, incrementally written output files in the <JOB>.output/ directory. The default setting addresses most of the issues common for large binary files: i.e., risk of data corruption, incremental downloads, and ease of archiving.

3.7.8.24 Convergence strategy (CONVERGENCE_STRATEGY)

Note:

Only available if Show advanced is checked, and if the field already exists in the file.

This setting is saved in the scenario file as CONVERGENCE_STRATEGY. This parameter controls activation of the rewind-restart feature

- automatic rewind and restart of the simulation with reduced time step in case of convergence errors (‘LARGE MASS RESIDUAL’ condition). By default, the rewind-restart feature is enabled - CONVERGENCE_STRATEGY=1.

3.7.9 Time dependent CFL numbers

This section in the scenario menu allows experienced users to prescribe CFL numbers that vary during the course of a simulation. The values are written to a cn-file for the current job number.

See also:

Time dependent CFL file

3.7.10 Dump/load settings

A runtime simulation control file (cc-file) can be written from CASD using the Dump/load settings section. To open this section of the scenario menu, right-click in the scenario menu, then select Dump/load settings.

![Figure 3.21: Right-click into the scenario menu to activate the Dump/load settings section.](image)

In the left column of the editor, you can insert cc-file keywords using a drop-down menu. In the right column, the corresponding numerical value is set. For adding or removing rows from the cc-file, right click
in the dump/load settings editor, then select Add row or Delete last row. Note that NDUMP must not be set to 8887 or 8888 since these are reserved dumpids that are used for creating restart data dumps.

Figure 3.22: Left: to add or delete rows right-click in the dump/load settings menu; right: the relevant command can be chosen from a drop-down menu, and the associated value entered in the right column.

Attention:

When saving the scenario, the cc-file is written only if the Dump/load settings menu is enabled. If the scenario is saved while the Dump/load settings menu is disabled, then any existing cc-file is deleted!

3.7.11 Boundary conditions

In the Boundary condition menu, you must specify boundary conditions for the outer boundaries of the simulation domain. The lower boundaries in X-, Y-, and Z-direction are denoted by XLO, YLO and ZLO respectively, and the upper boundaries likewise by XHI, YHI, ZHI. Recommended boundary conditions are as follows:

**EULER:** Euler equations

**NOZZLE:** Nozzle formulation

**PLANE, WAVE:** Plane wave condition

**WIND:** Wind inflow or outflow

The SYMMETRY boundary condition is generally not used for realistic geometries. In addition, it is also possible to choose EQCHAR, and BERNOULLI, but these boundary conditions are generally not recommended.

Remarks:

- For most explosion simulations, EULER can and should be used.
- For wind and dispersion simulations, the NOZZLE boundary condition (similar to EULER) is more robust.
- The PLANE, WAVE boundary condition is recommended for explosions in low confinement and for far field blast propagation. The boundary must be positioned far away from the explosion (flames should not reach the boundary and the domain should be large in comparison to the expanded cloud size).
- For modelling the surface of the sea or ground, it is recommended to use a solid surface together with either the EULER or the NOZZLE boundary condition.
- Different boundaries do not need to have the same condition.
- Boundary conditions try to model what happens beyond the boundary. Except for solid walls, this is not straightforward. Sometimes the boundary condition will disturb or even destroy a simulation. Then you should:
1. Ensure that the chosen boundary conditions are appropriate for the problem.
2. Consider increasing the Simulation volume and moving the boundaries to regions where the gradients that cross the boundaries are shallower.

Table 3.23: Recommended boundary conditions per simulation type.

<table>
<thead>
<tr>
<th>dispersion/ventilation</th>
<th>explosion: confined</th>
<th>explosion: unconfined/far-field</th>
</tr>
</thead>
<tbody>
<tr>
<td>EULER/NOZZLE + solid surface</td>
<td>ground/sea</td>
<td>ground/sea</td>
</tr>
<tr>
<td>EULER -</td>
<td>all other</td>
<td></td>
</tr>
<tr>
<td>NOZZLE - outflow</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>PLANE_WAVE -</td>
<td>- all other</td>
<td></td>
</tr>
<tr>
<td>WIND inflow/parallel boundaries</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

The details of various boundary conditions are given below.

3.7.11.1 Euler

The inviscid flow equations (Euler equations) are discretised for a boundary element. This means that the momentum and continuity equations are solved on the boundary in the case of outflow. The ambient pressure is used as the pressure outside the boundary. A nozzle formulation is used in the case of inflow or sonic outflow.

Warning:

The EULER boundary condition may give too low explosion pressures in unconfined situations due to reflections from the boundary. In such cases, the Simulation volume should be extended and the Plane wave boundary condition should be applied.

3.7.11.2 Nozzle

A nozzle formulation is used for both sub-sonic inflow and outflow and sonic outflow. This condition is suitable for porous areas with small sharp edged holes or grids (e.g. louvres and gratings). A discharge coefficient is calculated from the area porosity and a drag coefficient. NOZZLE condition has shown to give a bit higher explosion pressures than EULER, but it is more robust.

Warning:

The NOZZLE boundary condition may give too low explosion pressures in unconfined situations. In such cases, the Simulation volume should be extended and the Plane wave boundary condition should be applied.

3.7.11.3 Plane wave

This boundary condition was designed to reduce the reflection of the pressure waves at open boundaries that occurs when using EULER or NOZZLE. The pressure wave reflection is caused by setting a fixed pressure at the boundary. The PLANE_WAVE boundary condition extrapolates the pressure in such a way that reflections are almost eliminated for outgoing waves. However, the outflow is more restricted than for the EULER or NOZZLE conditions.
Attention:

The pressure may stabilise at a slightly elevated level after an explosion. For low confinement scenarios, it is recommended to use the PLANE_WAVE boundary condition and it is important that the grid recommendations are followed to ensure that there is sufficient distance between the total simulation boundary and the combustion region (or the high pressure region for blast scenarios) to avoid any unphysical boundary effects.

Warning:

In semi-confined situations where the boundaries are close to the vents, PLANE_WAVE should not be applied. PLANE_WAVE boundary condition requires a large simulation volume to work and flames should never reach the boundary.

3.7.11.4 Wind

The WIND boundary condition models an external wind field. Velocity and turbulence profiles are specified at the wind boundaries, either by setting some turbulence parameters manually or by choosing one of the atmospheric stability classes, see Pasquill class. The WIND boundary condition is particularly applicable to dispersion scenarios. It is possible to apply WIND both on inflow boundaries and on boundaries where the flow is parallel to the boundary.

Warning:

In cases where a generated internal flow has a strong impact on the boundary flow, e.g. gas explosions, WIND should not be used.

![Figure 3.23: Specification of Wind boundary condition](image)

3.7.11.4.1 Wind wizard

The wind wizard can be used to set up the boundary conditions if one or more of the boundary conditions are wind. The wizard can be launched from the Run Wizard->Wind wizard button available in the Scenario Settings section.
The wind wizard will set the boundary conditions according to the guidelines.

3.7.11.4.2 Wind speed  \( W_{\text{IND \_SPEED}} \), \( U_0 \), is the velocity on the boundary at a given Reference height. It is possible to set \( W_{\text{IND \_SPEED}} \) to positive, zero or negative values, but Gexcon recommends to set a positive value and use the Wind direction parameter to specify the direction of the wind. In case of no wind, consider to use another boundary condition. A uniform velocity profile is obtained by setting the Reference height equal to zero. Then the total volumetric flux over the boundary is as follows:
\[ V = U_0 \sum_n A_n \beta_n^2 \]  

(3.2)

### 3.7.11.4.3 Wind direction

WIND\_DIRECTION is a vector and each component may be given a positive, zero or negative value. The sign of this parameter determines the flow direction. A positive value means wind flow in positive direction, that is inflow over the lower boundaries and outflow at the upper boundaries. Wind at an angle different from axis directions may be specified using the WIND\_DIRECTION vector.

### 3.7.11.4.4 Relative turbulence intensity

RELATIVE\_TURBULENCE\_INTENSITY, \( I_T \), is the ratio between the isotropic fluctuating velocity, \( u' \), and the mean flow velocity \( U_0 \):

\[ I_T = \frac{u'}{U_0} \]  

(3.3)

\( I_T \) is used to calculate the value for turbulent kinetic energy, \( k = \frac{3}{2}u'^2 \), at the boundary.

**Attention:**

It is not necessary to set RELATIVE\_TURBULENCE\_INTENSITY for inflow boundaries when a Pasquill class is specified. When a Pasquill class is set, FLACS\-CFD will automatically create profiles for velocities and turbulence parameters at the boundary.

### 3.7.11.4.5 Turbulence length scale

TURBULENCE\_LENGTH\_SCALE, \( \ell_{LT} \) is a typical length scale at the boundary. It is used to calculate the rate of dissipation of turbulent kinetic energy at the boundary, \( \varepsilon \):

\[ \varepsilon = \frac{C_\mu k^{3/2}}{\ell_{LT}} \]  

(3.4)

It is important that the turbulence length scale is not assigned too high a value. As a guide, it should not exceed 10% of the smallest gas cloud dimension or 50% of the grid cell size. A turbulence length scale is not required when a Pasquill class is set.

### 3.7.11.4.6 Wind buildup time

WIND\_BUILDUP\_TIME is the time used to increase velocities on the boundaries from zero to WIND\_SPEED. A value for WIND\_BUILDUP\_TIME larger than zero gives a smooth start of the simulation. It is recommended to use WIND\_BUILDUP\_TIME for simulations with the incompressible solver, which may depend on the increased numerical stability obtained by a gradual increase of the wind speed in the initial phase. This is normally not a problem with the compressible solver. When the steady-state solver is enabled, WIND\_BUILDUP\_TIME is ignored and the velocity field is initialized with no gradual buildup.

### 3.7.11.4.7 Best practice

- Significant distance from the geometry to the simulation boundary is recommended (2-3 times the dimension of the facility)
- WIND on parallel and inflow boundaries and NOZZLE for outflow (however, use NOZZLE for +Z boundary in case of hot air releases to allow for outflow)
- Make sure that NOZZLE boundary is not too near, otherwise the wind will get diverted and find the easiest way out. A typical grid spacing may be 1-1.5 m, with stretching to enlarge cells away from the facility
- the EULER or NOZZLE boundary condition should be used for the ground/sea level, together with a solid surface.
3.7 Scenario menu

3.7.11.5 Fluctuating wind

Using a setup file, you can also specify a fluctuating wind field as boundary condition. By using fluctuating wind fields it is possible to mimic time-varying wind conditions. Two different frequencies in each horizontal direction and one in the vertical direction can be used:

```
VERSION 1.1
$WINDGUST
   USE=.TRUE.
$END
```

To change the frequencies of the fluctuations, more options must be specified:

```
VERSION 1.1
$WINDGUST
   USE = .TRUE.
   AMP = 2.40, 1.90, 1.84
       , 2.40, 1.90, 0.00
   TAU = 15.0, 10.0, 10.0
       , 70.0, 50.0, 0.0
$END
```

Two different fluctuating periods are assumed along (15 s and 70 s) and across (10 s and 50 s) the wind direction. In the vertical direction a single period of 10 s is used. Fluctuations are done as harmonic periods with average velocity fluctuation equal to 2.4 (along), 1.9 (across) and 1.3 (vertical direction; here the constant is multiplied by $\sqrt{2}$ since only one period is used) times the friction velocity, $u^*$. Remarks:

Except for in model validation studies, fluctuating wind fields are not used very often.

3.7.11.6 Using TRACER mode when simulating dispersion

In 'tracer' mode Flacs will only solve a passive transport equation. This option can be useful when a dispersion of neutral gas (or small quantities of gas) shall be simulated in an established wind field. Below is an example of how to enable the tracer mode using the Setup file:

```
VERSION 1.1
$SETUP
   KEYS="TRACE=T:100,DT_MUL=Y:5"
$END
```

The above setup file (or the KEYS defined in the scenario file) will simulate normally until time=100 s, thereafter only the fuel transport equation will be solved (flow field will be kept constant). When the flow field equations are switched off at 100 s, the time step is at the same time increased by a factor of 5 if using the DT_MUL=Y:5 string.

Warning:

This option should be used with care!

3.7.11.7 Effect of temperature gradients

To simulate the effect of an inversion layer, it is possible to define a cold or hot temperature region (layer) in FLACS-CFD. One can for instance define a cold valley by using the setup file as described in previous sections. Boundary conditions can not take a temperature profile.

3.7.11.8 General considerations for boundary conditions

It is generally advantageous to place the outer boundaries of the simulation domain far away from the geometrical extent, but limitations of memory and computing speed may restrict the practical size of the problem, and in most cases one is forced to compromise between quality and cost.
3.7.11.8.1 Solid wall boundary  The solid wall boundary condition is straightforward to model. The velocity vectors are zero at solid walls, both in the tangential and the perpendicular directions. Hence, a zero gradient perpendicular to the boundary, or a fixed value, works well for the scalar variables. Furthermore, wall-functions may improve the modelling of the flow in near-wall regions, both at the outer boundaries and in the interior space.

3.7.11.8.2 External influences  Obstacles that are outside the vent openings of a semi-confined volume should be included in the total simulation volume because they may have an effect on the explosion. One effect may be that the total venting is reduced due to the external obstacles, especially if they are large and are placed close to the vents. Since the vent flow is changed, also the internal flow past obstacles is modified and the explosion becomes different (higher or lower pressure results). An effect which also may be important is the appearance of an external explosion which will start when the flame reaches any unburnt gas which may have escaped through the vent openings. The pressure waves from the external explosion will propagate back into the semi-confined volume and give rise to higher pressures there. The strength of the external explosion will depend on the local turbulence in the external space, this again depends on the properties of the vent openings and on any obstacles which may be positioned in the external space.

3.7.11.9 Boundary condition at the sea (or other liquid) surface during an explosion

When simulating an explosion above the sea, the sea surface is generally modelled as a rigid, totally blocked surface (a solid plate with area porosity equal to zero). In the real, physical world, it is likely that the water surface would yield to some extent when hit by a blast wave (despite the water itself being close to incompressible). However, the sea surface is generally expected to move only a limited distance during the time frame of a blast wave hitting it (period of pressure pulse is typically of the order of 1-2 hundred milliseconds), and this is unlikely to be sufficient to significantly affect the simulated overpressure in the central region of the explosion (assuming that there is some distance between the sea surface and the central region). Similar arguments can be made for other liquid-gas interfaces.

3.7.11.10 Boundary conditions for explosion after dispersion

The following approach is recommended when running an explosion simulation based on dispersion dump file.

- Do not stop the release before ignition
- Keep the boundary conditions
- Make sure the boundaries are sufficiently far away so that the explosion can develop properly before blast waves hit the boundary.

There may be concern about reflections of pressure from WIND boundaries, however at the time the pressure waves hit the boundaries one should be careful trusting the continued pressure curves also with EULER due to the strong negative reflected pressure sent back into the domain.

What one will experience with this suggested best practice is that everything is fine until pressure waves hit and get reflected from the WIND boundary. If the boundary conditions are changed one will see unphysical behaviour immediately when the explosion simulation starts.

3.7.12 Initial conditions

Initial conditions set values for turbulence fields, temperature and pressure at the beginning of the simulation. Information about the gravity conditions, parameters for the atmospheric boundary layer and the composition of the air is also set here.

The default values are as follows:
3.7 Scenario menu

UP-DIRECTION  0  0  1
GRAVITY_CONSTANT  9.8
CHARACTERISTIC_VELOCITY  200.0
RELATIVE_TURBULENCE_INTENSITY  0.3
TURBULENCE_LENGTH_SCALE  0.05
TEMPERATURE  20.0
AMBIENT PRESSURE  100000
AIR  "NORMAL"
GROUND_HEIGHT  0
GROUND_ROUGHNESS  0
REFERENCE_HEIGHT  0
LATITUDE  0
SURFACE_HEAT_P1  0  0  0
SURFACE_HEAT_P2  0  0  0
MEAN_SURFACE_HEAT_FLUX  0
PASQUILL_CLASS  "NONE"

Attention:

Gexcon recommends that the default values are used for all explosion simulations where the appropriate values are not known.

3.7.12.1 Up-direction

This is a three-component vector defining the upward direction, i.e. opposite of the acceleration due to gravity. The three components denote the spatial components \(x, y, \) and \(z,\) respectively. It is possible to define any direction, not only directions aligned with the main axes. The vector does not need to be of unit length.

3.7.12.2 Gravity constant

GRAVITY_CONSTANT is the magnitude of the gravitational acceleration, normally \(9.8 \text{ m/s}^2.\) Setting a zero value for this parameter means that there are no gravitational influences on the flow. Panels with inertia are also influenced by this parameter if the direction of panel release is in the direction of the Up-direction, see also Pressure relief panels.

3.7.12.3 Characteristic velocity

CHARACTERISTIC_VELOCITY, \(U_0,\) is the mean flow velocity, and is used to find values for initial turbulence fields. It should take a positive or a zero value, see also Initial turbulence for the equivalent gas cloud.

3.7.12.4 Relative turbulence intensity

RELATIVE_TURBULENCE_INTENSITY, \(I_T,\) is the ratio between the isotropic fluctuating velocity, \(u',\) and the mean flow velocity, \(U_0: \)

\[
I_T = \frac{u'}{U_0}
\]  \hspace{1cm} (3.5)

The initial value for RELATIVE_TURBULENCE_INTENSITY is used, together with Characteristic velocity, to calculate the turbulent kinetic energy, \(k = 3/2u'^2,\) at the beginning of the simulation, see Initial turbulence for the equivalent gas cloud.

Remarks:

If RELATIVE_TURBULENCE_INTENSITY is set to zero and a Pasquill class is provided, then the initial values for the turbulent kinetic energy, \(k,\) and its dissipation, \(\varepsilon,\) are set to be similar to their values at the wind boundaries.
3.7.12.5 Turbulence length scale

TURBULENCE_LENGTH_SCALE, \( \ell_{LT} \), is a typical length scale that is used to calculate the dissipation rate of the turbulent kinetic energy, \( \varepsilon \):

\[
\varepsilon = \frac{C_k k^{3/2}}{\ell_{LT}}
\]  

(3.6)

It is generally around half of the hydraulic diameter for internal flows. It is important that the initial value for the TURBULENCE_LENGTH_SCALE is not assigned too high a value. As a guide, for explosion simulations it should not exceed 10% of the smallest cloud dimension or 50% of the grid cell size, see Initial turbulence for the equivalent gas cloud.

When the initial values for all three parameters:

- CHARACTERISTIC VELOCITY,
- RELATIVE TURBULENCE INTENSITY, and
- TURBULENCE LENGTH SCALE,

are set to the 0 and no Pasquill class is provided, then the turbulent kinetic energy per unit mass, \( k \), is set to 0.001 \( m^2/s^2 \) and the dissipation rate for the turbulent kinetic energy, \( \varepsilon \), is set to 0.001 \( m^2/s^3 \) (these default values may be changed using a setup file).

If the initial values for all three parameters are set to positive values, then initial values for \( k \) and \( \varepsilon \) will be calculated and used in preference to the default values (if \( k \) and \( \varepsilon \) are calculated to be exactly zero, then FLACS-CFD will set them to a very small number to prevent instability).

3.7.12.6 Temperature

The initial temperature, \( T_0 \) in °C. The default value is \( T_0 = 20.0^\circ \)C

3.7.12.7 Ambient pressure

AMBIENT PRESSURE, \( P_0 \) is the initial pressure in the simulation and the pressure outside the simulation volume. The default ambient pressure is \( P_0 = 100000 \) Pa = 1 bar.

3.7.12.8 Air

AIR is used to define the composition of the air. "NORMAL" denotes a standard composition of 20.95% oxygen and 79.05% nitrogen in mole fractions. The composition of air can be changed by setting an other oxygen content, either as a mole fraction:

AIR "25%MOLE"

mass fraction:

AIR "10%MASS"

or volume fraction:

AIR "10%VOLUME"

Changing the air composition will influence the lower flammability limit and upper flammability limit.
3.7 Scenario menu

3.7.12.9 Wind boundary layer parameters

In the Wind boundary condition, the reference wind speed, $U_0$, and the direction of the wind are specified. In addition, information about the ground conditions is needed to set inlet profiles for velocity and turbulence parameters. It is possible to specify an atmospheric boundary layer stability class, Pasquill class. Then turbulence parameter profiles are generated at the wind boundaries. If no Pasquill class is given, uniform values for $k$ and $\varepsilon$ are obtained on the boundary according to the expressions in Relative turbulence intensity and Turbulence length scale.

The velocity profile at the wind boundaries is given by the following expression:

$$u(z) = \begin{cases} \frac{u^*}{\kappa} \left( \ln \left( \frac{z-z_d+z_0}{z_0} \right) \right) - \psi_u(z) & \text{if } z_0 > 0 \\ U_0 & \text{if } z_0 = 0 \end{cases}$$

where $z$ is the height above the ground, $z_0$ is the atmospheric roughness length, $z_d$ is the canopy height (displacement height) and $u^*$ is a friction velocity. For the neutral and none Pasquill class, $u^*$ is defined by:

$$u^* = \frac{U_0 \kappa}{\ln \left( \frac{z_{ref}-z_d+z_0}{z_0} \right)} - \psi_u(z_{ref})$$

$\psi_u$ equals zero for none and neutral stability class (Pasquill class D), see Wind boundary.

Attention:

In FLACS v9.1 and later versions, vertical velocity and turbulence profiles start at the ground or at a given height (canopy height) above the ground. This is different from FLACS v9.0 and earlier versions where the vertical profiles start at a given height (ground height).

3.7.12.9.1 Ground height

GROUND_HEIGHT is the absolute $z$-value where the boundary layer starts in FLACS v9.0 and earlier versions. Ground height is not used in the default model for the atmospheric boundary layer in FLACS v9.1 and later versions. The old wind boundary conditions can used by setting:

KEYS = "WIND_MODEL=2"

3.7.12.9.2 Pasquill class

Pasquill atmospheric stability classes is a method of categorising the amount of atmospheric turbulence present. Pasquill (1961) defined six stability classes where:

A is very unstable,
B is unstable,
C is slightly unstable,
D is neutral,
E is slightly stable,
F is stable.

An overview when to apply the different Pasquill classes is given in table Pasquill stability classes.

Table 3.24: The Pasquill stability classes

<table>
<thead>
<tr>
<th>Wind speed</th>
<th>Day, strong sun</th>
<th>Day, moderate sun</th>
<th>Night, clouds $&gt;50%$</th>
<th>Night, clouds $&lt;50%$</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 2 m/s</td>
<td>A</td>
<td>B</td>
<td>E</td>
<td>F</td>
</tr>
<tr>
<td>2 - 3 m/s</td>
<td>A-B</td>
<td>B-C</td>
<td>E</td>
<td>F</td>
</tr>
<tr>
<td>3 - 5 m/s</td>
<td>B</td>
<td>B-C</td>
<td>D</td>
<td>E</td>
</tr>
<tr>
<td>5 - 6 m/s</td>
<td>C</td>
<td>C-D</td>
<td>D</td>
<td>D</td>
</tr>
<tr>
<td>&gt; 6 m/s</td>
<td>C</td>
<td>D</td>
<td>D</td>
<td>D</td>
</tr>
</tbody>
</table>
Attention:

One should be careful while using Pasquill classes A-C since they are unstable, and can lead to stability problems with the simulations. The results may not be optimal when these stability classes are used and in some cases, uncontrolled turbulence “exploding” leaks is observed. Pasquill class D is recommended in these cases. They are also more sensitive to increased time step (CFL-numbers) than Pasquill classes D-F. It is recommended to choose Pasquill class D instead of A, B or C.

3.7.12.9.3 Ground roughness  GROUNDROUGHNESS, $z_0$, refers to the aerodynamic roughness length. Typical values for $z_0$ are given in table Typical values for aerodynamic roughness length. $z_0$ should not be larger than the control volume height close to the surface. The aerodynamic roughness length should not be mixed with pipe roughness etc., but a rule of thumb is to relate $z_0$ to the average height $\varepsilon_g$ of the surface irregularities by $z_0 = \varepsilon_g/30$.

<table>
<thead>
<tr>
<th>Terrain description</th>
<th>$z_0$ (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open water, fetch at least 5 km</td>
<td>0.0002</td>
</tr>
<tr>
<td>Mud flats, snow; no vegetation, no obstacles</td>
<td>0.005</td>
</tr>
<tr>
<td>Open flat terrain; grass, few isolated obstacles</td>
<td>0.03</td>
</tr>
<tr>
<td>Low crops; occasional large obstacles</td>
<td>0.10</td>
</tr>
<tr>
<td>High crops, scattered obstacles</td>
<td>0.25</td>
</tr>
<tr>
<td>Parkland, bushes, numerous obstacles</td>
<td>0.5</td>
</tr>
<tr>
<td>Regular large obstacle coverage (suburb, forest)</td>
<td>1.0</td>
</tr>
</tbody>
</table>

3.7.12.9.4 Reference height  REFERENCE_HEIGHT, $z_{ref}$ is the height relative to the ground where the velocity equals the wind speed.

3.7.12.9.5 Canopy height  CANOPY_HEIGHT, $z_d$, refers to the height above the ground where the boundary layer actually starts, for instance due to vegetation or buildings. Usually CANOPY_HEIGHT=0. When a value is set for the canopy height, it should be about 2/3 of typical tree or building size. The canopy height must be equal or larger than zero and less than the reference height.

3.7.12.9.6 Latitude  Due to the rotation of the earth, the height of the atmospheric boundary layer is much larger at equator than at the poles. LATITUDE will only have an effect if a Pasquill class is chosen and the simulation volume is very large (> 200 m) in the z direction.

3.7.12.9.7 Surface heat P1  If the temperature and velocity are known at two different altitudes, for instance from experimental data, it is possible to estimate the surface heat flux. SURFACE HEAT P1 is a vector $Z_1 U_1 T_1$, where $Z_1$ is the altitude, $U_1$ is the velocity at $Z_1$ and $T_1$ is the temperature at $Z_1$.

3.7.12.9.8 Surface heat P2  SURFACE HEAT P2 is a vector $Z_2 U_2 T_2$, where $Z_2$ is the altitude, $U_2$ is the velocity at $Z_2$ and $T_2$ is the temperature at $Z_2$. $Z_2$ must larger than $Z_1$ and $U_2$ must larger than $U_1$.

3.7.12.9.9 Mean surface heat flux  MEAN_SURFACE_HEAT_FLUX is the heat flux in $W/m^2$ from the ground to the flow. This is a parameter in the boundary layer profiles for the unstable Pasquill class A, B, and C. MEAN_SURFACE_HEAT_FLUX will not apply as heat contribution from the ground to the flow in the simulations.
3.7 Scenario menu

3.7.13 Gas composition and volume

This section allows you to define a box shaped cloud region and the gas concentration and composition. The menu is shown below:

![Figure 3.25: The scenario menu that allows to define the gas composition and volume.](image)

This menu corresponds to the scenario file section:

```
POSITION_OF_FUEL_REGION 0.0, 0.0, 0.0 (m)
DIMENSION_OF_FUEL_REGION 0.0, 0.0, 0.0 (m)
TOXIC_SPECIFICATION ""
VOLUME_FRACTIONS <MENU>
EQUIVALENCE_RATIOS (ER0_9) 1.0, 0.0
```

The position of the cloud is the location of the minimum point \((x_{\text{min}}, y_{\text{min}}, z_{\text{min}})\) of the box, and the sizes \((x_{\text{siz}}, y_{\text{siz}}, z_{\text{siz}})\) give the side lengths of the box (only positive values are allowed for the sizes). On expanding the Volume fractions entry, the gas composition can be specified by entering the volume fractions of the listed gas components. The concentration is given by the equivalence ratios (ER0 inside the gas cloud and ER9 outside).

Example of the cs-file for fuel consisting of methane only:

```
METHANE 1.0
TOXIC 0
```

The Gas composition and volume section has a button at the bottom named Edit volume fractions that opens a Volume fractions editor dialog which is used for modifying the selection of species in the volume fractions list. This dialog contains a list of "Available species" that can be added to a list of "Used species" by double clicking on an available specie, or by selecting a species and clicking on the right-arrow between the two lists. Species can be removed from the "Used species" list by selecting a specie and clicking on the left-arrow. The list of available species includes a "Properties" column, with icons that indicate if a species is flammable and/or toxic, and a "SLAM Data" column that uses a check-box to indicate if SLAM data is available. User species (if defined in the User species section) and the Toxic specie cannot be deleted from the list of used species. The volume fractions editor dialog is shown in the figure below.
Figure 3.26: The volume fractions editor dialog, used for modifying the list of species in the Volumes fractions section

Attention:

USERSPEC_1, USERSPEC_2, USERSPEC_3 can be used as custom user species. Properties are updated automatically when species are added, deleted or renamed in User species section. The volume fraction values need to be set by the user.

The sum of the volume fractions does not need to be 1. FLACS-CFD will interpret the values as volumetric parts (i.e. normalisation is done in FLACS-CFD by dividing each value by the sum of all values).

For a mixture of two or more gas components, the combustion properties are calculated as a weighted average of all the selected components, taking into account the mole (volume) fraction and the relative consumption of O2 (parameter a in the table above). This has shown to give the proper blending of the component properties into mixture properties.

The laminar burning velocity of a gas mixture in FLACS-CFD depends on the concentration of the fuel relative to the concentration of oxygen as well as on the type of fuel. A widely used measure for the relative fuel-oxygen concentration is the Equivalence ratio (ER) which is defined as follows:

\[
ER = \theta = \frac{(m_{\text{fuel}}/m_{\text{oxygen}})_{\text{actual}}}{(m_{\text{fuel}}/m_{\text{oxygen}})_{\text{stoichiometric}}} = \frac{(V_{\text{fuel}}/V_{\text{oxygen}})_{\text{actual}}}{(V_{\text{fuel}}/V_{\text{oxygen}})_{\text{stoichiometric}}} \quad (3.9)
\]

The dependency of the laminar burning velocity (Slam) is illustrated for different species as function of the equivalence ratio ER in the figure below.
3.7 Scenario menu

Figure 3.27: Laminar burning velocity for individual gas components. Hydrogen, acetylene and ethylene have the three highest laminar burning velocities, and notably wide ranges of flammability, too.

Attention:

The correlation in FLACS-CFD accounting for initial pressure effects on the flammability limits is not valid for pressures significantly below atmospheric values. For these particular cases, you can define your own gasdata file, if you have access to reliable experimental data. Similarly, check in the literature whether the properties of mixtures of different fuels are reasonable. A modified gasdata file might be used where unphysical results are observed.

Warning:

The stoichiometry model implemented in FLACS-CFD describes combustion reactions in which carbon, hydrogen and sulphur atoms combine with oxygen (available in air or in the fuel itself) leading to combustion products. Presence in the fuel of other atoms that can react with oxygen or, in general, with chemical elements from the atmosphere or from the fuel, such as fluorine, is not taken into account in the stoichiometry model implemented in FLACS-CFD. For further guidance on simulating combustion of non-hydrocarbons, please contact FLACS-CFD support.

3.7.13.1 Hydrogen-Nitrogen specific laminar burning velocity model

A special model for determining the laminar burning velocity (SLAM) and flammability limits LFL, UFL for hydrogen-nitrogen gas mixtures was added in FLACS v10.9. This model yields more accurate values for SLAM, LFL and UFL when nitrogen is present in the gas mixture and resolves an issue related to overprediction of LFL for hydrogen-nitrogen mixtures in previous versions of FLACS-CFD. The new model is automatically activated when the gas mixture contains only hydrogen and some fraction of nitrogen, and for all other gas mixtures the old model is used by default. It is possible to override the automatic activation by giving the key 'SLAMCORR_MODEL=1/2', where 1 = 'old' model and 2 = 'new' model. The new model uses a hydrogen-nitrogen specific lookup table and will therefore not work for any other mixtures. Note the new model has not yet been implemented in DustEx.

The simulation log file contains information about which model was activated: either

# SLAMCORR_MODEL = 1 (DEFAULT)
OR
# SLAMCORR_MODEL = 2 (HYDROGEN-N2)  [new model]

Attention:

When modelling hydrogen-nitrogen gas mixtures, it is very important to verify that the 'new' model has been used, by verifying that the logfile includes: "SLAMCORR_MODEL = 2 (HYDROGEN-N2)"
3.7.13.2 Toxic gas definition

It is possible to model the effect of toxic substances with FLACS-CFD. The toxic component is specified in the section GAS_COMPOSITION_AND_VOLUME in the scenario menu or in the corresponding section in the cs-file. A description of probit functions and probability of death related to toxic substances, is also found in Sec. 5.2 “Damage modelling” in TNO 'Purple book'.

The parameter TOXIC_SPECIFICATION can be specified in one of the following ways:

- selecting a predefined substance,
- creating a user-specified toxic data file and specifying the substance name,
- specifying the substance properties directly.

The following predefined substances are available:

Table 3.26: Predefined toxic substances

<table>
<thead>
<tr>
<th>Substance</th>
<th>a</th>
<th>b</th>
<th>n</th>
<th>Formula</th>
<th>Molar mass (g/mol)</th>
<th>Boiling point (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acrolein</td>
<td>-4.1</td>
<td>1</td>
<td>1</td>
<td>C3H4O</td>
<td>56.06</td>
<td>53</td>
</tr>
<tr>
<td>Acrylonitrile</td>
<td>-8.6</td>
<td>1</td>
<td>1.3</td>
<td>C3H3N</td>
<td>53.06</td>
<td>77</td>
</tr>
<tr>
<td>Allyl alcohol</td>
<td>-11.7</td>
<td>1</td>
<td>2</td>
<td>C3H6O</td>
<td>58.08</td>
<td>97</td>
</tr>
<tr>
<td>Ammonia</td>
<td>-15.6</td>
<td>1</td>
<td>2</td>
<td>NH3</td>
<td>17.0306</td>
<td>-33.34</td>
</tr>
<tr>
<td>Azinphos-methyl</td>
<td>-4.8</td>
<td>1</td>
<td>2</td>
<td>C10H12N3O3PS2</td>
<td>317.32</td>
<td>200</td>
</tr>
<tr>
<td>Bromine</td>
<td>-12.4</td>
<td>1</td>
<td>2</td>
<td>Br2</td>
<td>159.808</td>
<td>58.85</td>
</tr>
<tr>
<td>Carbon monoxide</td>
<td>-7.4</td>
<td>1</td>
<td>1</td>
<td>CO</td>
<td>28.010</td>
<td>-192</td>
</tr>
<tr>
<td>Chlorine</td>
<td>-6.35</td>
<td>0.5</td>
<td>2.75</td>
<td>Cl2</td>
<td>70.906</td>
<td>-34.4</td>
</tr>
<tr>
<td>Ethylene oxide</td>
<td>-6.8</td>
<td>1</td>
<td>1</td>
<td>C2H4O</td>
<td>44.05</td>
<td>10.7</td>
</tr>
<tr>
<td>Hydrogen chloride</td>
<td>-37.3</td>
<td>3.69</td>
<td>1</td>
<td>HCl</td>
<td>36.46</td>
<td>-85.1</td>
</tr>
<tr>
<td>Hydrogen cyanide</td>
<td>-9.8</td>
<td>1</td>
<td>2.4</td>
<td>HCN</td>
<td>27.03</td>
<td>26.0</td>
</tr>
<tr>
<td>Hydrogen fluoride</td>
<td>-8.4</td>
<td>1</td>
<td>1.5</td>
<td>HF</td>
<td>20.01</td>
<td>19.54</td>
</tr>
<tr>
<td>Hydrogen sulfide</td>
<td>-11.5</td>
<td>1</td>
<td>1.9</td>
<td>H2S</td>
<td>34.082</td>
<td>-60.28</td>
</tr>
<tr>
<td>Methyl bromide</td>
<td>-7.3</td>
<td>1</td>
<td>1.1</td>
<td>CH3Br</td>
<td>94.94</td>
<td>3.56</td>
</tr>
<tr>
<td>Methyl isocyanate</td>
<td>-1.2</td>
<td>1</td>
<td>0.7</td>
<td>C2H3NO</td>
<td>57.1</td>
<td>39.1</td>
</tr>
<tr>
<td>Nitrogen dioxide</td>
<td>-18.6</td>
<td>1</td>
<td>3.7</td>
<td>NO2</td>
<td>46.01</td>
<td>21.1</td>
</tr>
<tr>
<td>Parathion</td>
<td>-6.6</td>
<td>1</td>
<td>2</td>
<td>C10H14NO5PS</td>
<td>291.3</td>
<td>375</td>
</tr>
<tr>
<td>Phosgene</td>
<td>-10.6</td>
<td>2</td>
<td>1</td>
<td>CCl2O</td>
<td>98.92</td>
<td>8</td>
</tr>
<tr>
<td>Phosphamidon</td>
<td>-2.8</td>
<td>1</td>
<td>0.7</td>
<td>C10H19CINO5P</td>
<td>299.70</td>
<td>162</td>
</tr>
<tr>
<td>Phosphine</td>
<td>-6.8</td>
<td>1</td>
<td>2</td>
<td>PH3</td>
<td>34.00</td>
<td>-87.8</td>
</tr>
<tr>
<td>Sulphur dioxide</td>
<td>-19.2</td>
<td>1</td>
<td>2.4</td>
<td>SO2</td>
<td>64.07</td>
<td>-10</td>
</tr>
<tr>
<td>Tetraethyl lead</td>
<td>-9.8</td>
<td>1</td>
<td>2</td>
<td>C8H20Pb</td>
<td>323.44</td>
<td>84</td>
</tr>
</tbody>
</table>
You can specify one of these substances as the toxic component in the scenario file section:

```
GAS_COMPOSITION_AND_VOLUME:

GAS_COMPOSITION_AND_VOLUME
  POSITION_OF_FUEL_REGION  0  0  0
  DIMENSION_OF_FUEL_REGION  0  0  0
  TOXIC_SPECIFICATION  "Ammonia"
VOLUME_FRACTIONS
  TOXIC  1
EXIT VOLUME_FRACTIONS
EQUIVALENCE_RATIOS_(ER0_9)  1E+30  0
EXIT GAS_COMPOSITION_AND_VOLUME
```

For a pure toxic release, specify $ER0 = 1E+30$, although the toxic substance might not be combustible. This is not formally correct, but is a workaround and gives a mass fraction of one.

The alternative is to write a user-defined toxic data file (`./toxic.dat`):

```
: substance , a , b , n , formula , MolarMass(g/mol) , BoilingPoint(°C)
"Acrolein" , -4.1 , 1. , 1. , "C3H4O" , 56.06 , 53.
="Acrylaldehyde"
```

If you specify a toxic substance under the parameter `TOXIC_SPECIFICATION` without setting a `TOXIC` volume fraction, as for example:

```
VOLUME_FRACTIONS
  USERSPEC_1  1
  TOXIC  0
EXIT VOLUME_FRACTIONS
```

the entire defined gas composition is treated as the toxic substance and all toxic properties (e.g., TDOSE, lethality etc.) are calculated for the entire gas. This can be useful when for example doing pure toxic gas dispersion or toxic liquid evaporation studies, for which the regular toxic thermodynamic properties are not accurate enough.

Attention:

The first sign in the toxic data file must not be space or tab. Comments starts with #, ! or ’’ (space).

The scenario specification should then look as follows (in this case there is also a fraction of methane in the gas mixture):

```
GAS_COMPOSITION_AND_VOLUME:

GAS_COMPOSITION_AND_VOLUME
  POSITION_OF_FUEL_REGION  0  0  0
  DIMENSION_OF_FUEL_REGION  0  0  0
  TOXIC_SPECIFICATION  "Acrolein, data_file=./toxic.dat"
VOLUME_FRACTIONS
  METHANE  1
  TOXIC  9
EXIT VOLUME_FRACTIONS
EQUIVALENCE_RATIOS_(ER0_9)  1E+30  0
EXIT GAS_COMPOSITION_AND_VOLUME
```

The substance properties can be specified directly in the `GAS_COMPOSITION_AND_VOLUME` section:

```
TOXIC_SPECIFICATION  "probit_constants=-4.1,1,1,molar_mass=56.06"
```

The keywords for the `TOXIC_SPECIFICATION` are:

- `substance` name of the toxic substance (or formula)
- `probit_constants` =a, b, n. See the explanation in the section on the Toxic probit function `PROBIT` in the description of the Output variables for toxic substances.
- `molar_mass` =M
**data_file** name of the datafile (default toxic_data.dat)

The **TOXIC_SPECIFICATION** string is stripped of spaces and converted to lowercase before parsing. The name of the datafile and formula are case sensitive.

Note that when a fuel component, e.g. H2S, is specified as a toxic species (**TOXIC_SPECIFICATION "H2S"**), then the volume (mole) fraction of H2S should be specified for **TOXIC**, not H2S. Thus, if for example the volume fraction of H2S as part of the fuel composition is 0.01 (1.0%), then **TOXIC 0.01 and H2S 0** should be specified as part of the parameters defined in **VOLUME_FRACTIONS** (rather than **TOXIC 0 and H2S 0.01**). The same applies to other toxic fuel components, like CO.

Toxic species are modelled by changing the molecular weight of a general “representative” inert species (CO2) whose combustion parameters are not necessarily correct, especially for flammable toxic components. Therefore, when combustion parameters, like the flammability limits, are important, it is recommended to avoid using the model for toxic species. Similarly, since thermodynamic properties are not individually defined for each species, simulation with toxics must be limited to scenarios with no temperature differences (i.e. releases at ambient temperature and no hot/cold solid objects), or scenarios characterized by temperature differences but low toxic concentration.

**Warning:**

The implemented models for toxic components are limited to substances with purely gaseous behaviour. Toxic substances with a boiling point above ambient temperature will typically spread as a mist and their toxic effect could for instance require direct skin contact. Such effects are currently not handled by FLACS-CFD.

**Attention:**

A gas cloud with toxic components should not be specified in the **GAS_COMPOSITION_AND_VOLUME** scenario section. To specify a given mass fraction in a predefined gas cloud use the cloud interface. Toxic specification cannot be used in combination with the fire solver and will give an error message when trying to run.

### 3.7.13.3 Inert gas definition (deprecated)

**Attention:**

This chapter is deprecated as FLACS_CFD no longer includes a separate inert solver.

Nitrogen and CO2 are included as default (inert) fuel species in most variants of Flacs (except DustEx). Other inert gasses can also be included by defining a user species. For cases where a separate inert gas region needs to be defined, this can be done in the special inert variant of Flacs. To use this model select 'Simulation type' Inert.

**Attention:**

The correlations in FLACS-CFD accounting for the effect of having an inerted atmosphere (oxygen content less than standard atmospheric content), are not validated for all possible mixtures and combinations of added inert gases. For the alkanes, the validation shows good agreement with the available literature. However, the lower flammability limit of hydrogen inerted with CO2 or H2O may be over-predicted. Treat these mixtures with care, and possibly apply a modified gasdata file if unphysical results are observed.

The **GAS_COMPOSITION_AND_VOLUME** menu has been extended to allow definition of **FUEL** and **INERT** regions (or clouds). This enables to specify shaped (rectangular or rounded) clouds, to specify the **FUEL** and **INERT** gas compositions, and to specify **FUEL** and **INERT** gas concentrations.
It is possible to define one FUEL region and one INERT region. The position, size and shape of the regions can be defined, as well as the composition and concentration of gases. Take care to understand the mixing rules for the inert/fuel/air and the combination of regions.

The shape of the region defines if the region is rectangular (square), elliptical (rounded) or otherwise shaped (convex or concave). The shape parameters must be given for the three main directions (x,y,z):

<table>
<thead>
<tr>
<th>Id</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>box</td>
</tr>
<tr>
<td>1</td>
<td>diamond</td>
</tr>
<tr>
<td>2</td>
<td>elliptical (similar to the sphere equation: (x^2 + y^2 + z^2 = r^2))</td>
</tr>
<tr>
<td>d/e/f</td>
<td>other shape (general ellipsoid equation: ((x/a)^d + (y/b)^e + (z/c)^f = 1))</td>
</tr>
</tbody>
</table>

A sphere has shape (2,2,2), a cylinder in the z-direction has shape (2,2,0), and a diamond has the shape (1,1,1). The larger the exponents are the closer to a box shape one comes, and the smaller the exponents are the closer to a cross one comes (except from 0 which gives a box).

The fuel composition is given by the volume fractions of species. The sum does not need to be 1, because the FLACS-CFD simulator will ensure this after reading the scenario file. Negative volume fractions are not permitted.

The most common hydrocarbon gases plus \(H_2\), \(CO\), \(H_2S\) and \(CO_2\) are available as fuel species. When using the FLACS-CFD inert application one should be aware that \(CO_2\) in the fuel is accounted for as an integrated part of the fuel, and is thereby a separate entity from the amount of \(CO_2\) in the inert.

The inert composition is given by the volume fractions of species. The sum does not need to be 1, because the FLACS-CFD simulator will ensure this after reading the scenario file. Negative volume fractions are not permitted.

Currently \(N_2\) and \(CO_2\) are available as inert species.

The VOLUME_FRACTIONS_INERT consist of 3 values where you enter the volume fraction for 3 of the 4 logical regions which are given by the combination of the FUEL and INERT regions (4 logical combinations of inside and outside the two exist):
Table 3.28: Inert gas regions

<table>
<thead>
<tr>
<th>Value</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INERT</td>
<td>inside INERT and outside FUEL region</td>
</tr>
<tr>
<td>2</td>
<td>FUEL</td>
<td>inside FUEL and outside INERT region</td>
</tr>
<tr>
<td>3</td>
<td>BOTH</td>
<td>inside FUEL and inside INERT region (the overlap region)</td>
</tr>
<tr>
<td>4</td>
<td>NONE</td>
<td>outside both FUEL and INERT (here ER9 gives the FUEL concentration)</td>
</tr>
</tbody>
</table>

By this means it is possible to make for example an inert cloud surrounding the fuel cloud. The values given are interpreted according to the mixing rules, they can be absolute values (=) or additive (+/-) or some other interpretation (mainly used for testing).

The figures below show some possible locations of the INERT and FUEL regions to illustrate the concept.

The mixing rule defines how the INERT is mixed with FUEL and AIR, the following key-words are recognised:

Table 3.29: Inert mixing rule keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>set value as final fraction</td>
</tr>
<tr>
<td>+</td>
<td>add value</td>
</tr>
<tr>
<td>-</td>
<td>subtract value</td>
</tr>
<tr>
<td>=N2inAIR</td>
<td>use $N_2$ content given for INERT as equivalent $N_2$ in AIR</td>
</tr>
<tr>
<td>=CO2inFUEL</td>
<td>use $CO_2$ content given for INERT as equivalent $CO_2$ in FUEL</td>
</tr>
</tbody>
</table>

The RULE_MIX_INERT is a text string that contains the mixing rule for 3 of the 4 logical regions which are given by the combination of the FUEL and INERT regions (4 logical combinations of inside and outside the two exist, see illustration for how regions are combined). The text string should contain commas (1,2,3) separating the key-word for each of the regions: only INERT (1), only FUEL (2) and BOTH (3).
3.7 Scenario menu

3.7.13.3.1 Start the Flacs inert simulator

The inert models are currently implemented in a special variant of the Flacs simulator, which is installed alongside the standard simulator. To start an inert simulation...
set the version by clicking the Parameters button in the RunManager, and set the FLACS-CFD version to "2.6.0inert".

To start a simulation on the command line:

Linux:

> run flacs version 2.6.0inert 010101

Windows:

> flacs version 2.6.0inert 010101

### 3.7.14 Multiple or custom gas clouds

One or several gas clouds can be defined for a Flacs simulation, either by specifying the characteristic data for some predefined cloud shapes, by entering the relevant data describing the (arbitrarily shaped) cloud directly in the scenario menu, or by pointing to an existing cloud file. These three ways will be described in the following.

#### 3.7.14.1 Using predefined cloud shapes

Right-click into the main field of the Multiple or custom gas clouds scenario menu and choose Create cloud.

![Image](image.png)

**Figure 3.33:** Left: The Multiple or custom gas clouds section in the scenario menu after right-clicking. Right: The Create new gas cloud dialogue box.

In the Create new gas cloud dialogue box, the shape can be selected from a drop-down box as one of the following: box, ellipsoid, cylinder, or custom. These are described below with their respective input parameters. All shapes require a position, which specifies the lower left front corner of the bounding box of the cloud (closest to the origin), and rotation angles with respect to the coordinate axes. Further, a subdivision parameter has to be specified; it determines the number of planar surfaces used to approximate curved shapes.

**• Box:** A box-shaped cloud is defined by its extent along the three (possibly rotated) coordinate axes and a constant mass fraction.

**• Ellipsoid:** Ellipsoids with different radii along the three coordinate axes can be defined. Also here, the fuel mass fraction is constant.

**• Cylinder:** A cylindrical gas cloud can be defined by its length (along the unrotated z-axis) and two radii (in the x- and y-directions). The mass fraction can be specified for both ends (low and high z-plane) and will be linearly interpolated between these values.
• **Custom**: The custom choice enables you to specify a cloud as the convex hull of a set of points that are generated in a parametric manner. To this end, the range and the number of steps can be given for three parameters \( i \), \( j \), and \( k \), which can then be used in three independent formulae to generate the \( x \), \( y \), and \( z \) coordinates of the points at which a value for the fuel mixture fraction is specified whose computation may likewise use the parameters \( i \), \( j \), and \( k \). The formulae to compute the coordinates and mixture fractions may use python expressions, e.g. mathematical functions from the math library as in \( \text{math.sin}(i)+2\cdot j \).

![Figure 3.34: Example of a custom cloud in the shape of a half cylinder and how it was defined.](image)

3.7.14.2 Specifying arbitrary cloud shapes

For more complicated shapes, or when the fuel mass fraction varies in a more general way over the volume than allowed by the predefined cloud shapes, CASD allows to define the cloud by a list of points with the fuel mass fraction given in each point. By right-clicking under the list of points, more lines can be added.

3.7.14.3 Including an existing cloud file

A *cloud file* can be specified in the Multiple or custom gas clouds part of the scenario menu. Details about the parameters and format of a cloud file can be found in the *cloud file* section.

3.7.15 Pool

The Pool section of the scenario menu defines the properties of a liquid spill on the ground or on water. When a pool is defined in CASD, a *POOL* section is added to the scenario file that is read by the simulator, and the pool is visualized as a semi-transparent disk. An example of a scenario file for a pool simulation (cs600000.dat3) is provided with the FLACS-CFD installation in the doc/examples/ex02.pool folder under the installation directory.

The models used for pool calculations are described in detail in the *Pool model* section. It is possible to derive a static pool from the default moving spill model. For a moving spill, the *shallow-water* equations are solved in two dimensions on the ground. Heat from the ground, the flow above the pool/spill, and the
radiation from the sun and/or fire determine the evaporation rate.

The composition of the liquid is read from the Gas composition and volume section of the scenario, so for a pool scenario, it is important to set:

```
EQUIVALENCE_RATIOS_1 (ER0_ER9) 1E+30, 0.0
```

The parameters that can be entered in this section in CASD are described in the following subsections.

3.7.15.1 Static pool (STATIC_POOL)

This flag specifies whether a static or spreading pool will be simulated. If the box is checked, then STATIC_POOL is set to one in the scenario file and a static pool will be simulated.

3.7.15.2 Pool shape (POOL_SHAPE)

At present, only circular pools can be simulated. This field is written to the scenario file as POOL_SHAPE.

3.7.15.3 Start time (START_POOL)

This is the time in the simulation from which the pool is present, and the value is written to the scenario file as the value for START_POOL. If START_POOL is set to zero, then the pool will be assumed present from the beginning of the scenario.

3.7.15.4 Mass (MASS_POOL_0)

This is the initial mass of the pool, before any evaporation or combustion, and is written to the scenario file as MASS_POOL_0.

3.7.15.5 DMDT (DMDT)

This is the rate of mass increase for the pool for a non-instantaneous spill. If entered here, then it is assumed to be constant for the duration of the spill and is distributed uniformly over the pool area. It is also possible to define a time-varying spill using an additional file, see Pool leakage file for details of this.

3.7.15.6 Position (POSITION)

The X, Y and Z coordinates for the center of the pool (for a static pool), or of the leak area (for a spreading pool). The Z coordinate is the altitude from which Flacs searches downwards to determine the vertical position of the pool base. The pool is positioned on the highest solid (or water) surface that is directly below Z. The Pool spread simulation example in the FLACS-CFD Best practice Section demonstrates how this process works.

3.7.15.7 Inner radius and Outer radius (RAD_I and RAD_O)

The values entered for the inner radius and outer radius are written to the scenario file as RAD_I and RAD_O, respectively. The inner radius defines the inner radius of a donut-shaped pool (for a static pool) or leak area (for a moving pool), and is set to zero by default. The outer radius is the outer radius of the pool or leak area.
3.7 Scenario menu

3.7.15.8 Ground temperature (T_SOIL)

The initial temperature of the substrate beneath the pool, i.e., it's temperature before the pool was present. This is written to the scenario file as T_SOIL. Note that for a water substrate, this parameter refers to the surface temperature of the water prior to the spill. If set to a value greater than zero, or not specified, then the ambient air temperature is used.

3.7.15.9 Heat sun (HEAT_SUN)

The radiative heat from the sun, which is written to the scenario file as HEAT_SUN. If not specified, a default value of 400 W m\(^{-2}\) is used.

3.7.15.10 Surface roughness (ROUGH_L)

The surface roughness of the liquid pool surface. If set to zero, then the default value of 0.005 m is used. This is written to the scenario file as ROUGH_L.

3.7.15.11 Ground type (POOL_GROUND)

The properties of the substrate beneath the pool are read according to the type of ground that is specified here, which is written to the scenario file as POOL_GROUND. The dropdown menu lists the available types: concrete, average, soil, plate, insulated, user and water. The thermal properties that are assumed for these types are detailed in the table below. If the ground type is specified as 'user', then the ground conductivity and diffusivity must be provided by the user. It is possible to specify more than two ground types for the substrate by manually entering text into the dropdown box (the text is then copied exactly to the scenario file). An example of how to do this is provided in Pool setup parameters in the Pool spread simulation example. The different ground types should be entered as a comma-separated list, followed by a space and then square brackets that contain a space-separated list of pairs of numbers, which specify the X, Y and Z bounds of the region over which the second ground type should be used.

<table>
<thead>
<tr>
<th>Ground</th>
<th>Conductivity ( Wm(^{-1})K(^{-1}))</th>
<th>Thermal diffusivity ( m(^2)s(^{-1}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concrete</td>
<td>1.1</td>
<td>10(^{-6})</td>
</tr>
<tr>
<td>Average/Soil</td>
<td>0.9</td>
<td>4.3 \cdot 10(^{-7})</td>
</tr>
<tr>
<td>Plate</td>
<td>15</td>
<td>3.9 \cdot 10(^{-6})</td>
</tr>
<tr>
<td>Insulated</td>
<td>0.0</td>
<td>10(^{30})</td>
</tr>
<tr>
<td>User</td>
<td>Set CONDUCT_S</td>
<td>Set DIFFUS_S</td>
</tr>
<tr>
<td>Water</td>
<td>Heat transfer coefficient</td>
<td>Heat transfer coefficient</td>
</tr>
</tbody>
</table>

3.7.15.12 Ground thermal conductivity (CONDUC_S)

Ground thermal conductivity. This field is only available in the Pool menu when Ground type is set to "User", and is written to the scenario file as CONDUC_S. If it is not specified, or is specified as zero, then the simulator will use the value for the ground type "Average/Soil" from the table.
3.7.15.13 DIFFUS_S (DIFFUS_S)

Ground thermal diffusivity. This field is only available in the Pool menu when "show advanced" is checked. It is required if the Ground type is set to "User". If Ground type is set to "User" and DIFFUS_S is not specified, or is specified as zero, then the simulator will use the value for the ground type "Average/Soil" from the table. For other ground types, the simulator will use this value if it is non-zero, but will otherwise use the appropriate value from the table.

3.7.15.14 Air thermal conductivity (CONDUCT_A)

Air thermal conductivity. This field is only available in the Pool menu when the Ground type is set to "User", and is written to the scenario file as CONDUCT_A. If it is not specified, or is specified as zero, then a default value of 0.024 W m⁻¹ K⁻¹ is used.

3.7.15.15 Plate thickness (PLATE_L)

The thickness of the steel plate. This field is only available in the Pool menu when the Ground type is set to "Plate", and is written to the scenario file as PLATE_L. If it is not specified, or is specified as 0, and Ground type is set to "Plate", then a default value of 10⁻³ m is used.

3.7.15.16 Albedo (ALBEDO)

The albedo of the pool liquid. Only available when "show advanced" is checked. If it is not specified, or is specified as zero, then a default value of 0.11 is used.

3.7.15.17 C_POOL (C_POOL)

The specific heat capacity of the pool liquid. Only available when "show advanced" is checked. If it is not specified, or if Use database is checked, then the simulator will calculate the appropriate value from the properties in the FLACSCFD database for the pool liquid at the initial pool temperature.

3.7.15.18 CFL_POOL (CFL_POOL)

CFL number to be used for calculations in the pool model. Only available when "show advanced" is checked. If it is not specified, or is specified as zero, then a default value of 0.8 is used.

3.7.15.19 DADT (DADT)

Rate of expansion of the pool area. Only available when "show advanced" is checked.

3.7.15.20 Use database (DATABASE)

This flag specifies whether the properties of the pool should be read from the FLACSCFD database, or be specified here by the user. Only available if "show advanced" is checked. If the box is checked, then DATABASE is set to one in the scenario file and the FLACSCFD database is used. This is the default behaviour.
3.7.15.21 EMISSION_A and EMISSION_P (EMISSION_A and EMISSION_P)

Emission coefficients for the air (EMISSION_A) and pool (EMISSION_P). Only available if "show advanced" is checked. If these are not specified, or are set to values that are not between zero and one, then default values of EMISSION_A = 0.75 and EMISSION_P = 0.95 are used.

3.7.15.22 H_WATER (H_WATER)

Heat transfer coefficient between water and the pool. Only available if "show advanced" is checked. If it is not specified, then a default value of 0 is used.

3.7.15.23 HEAT_EVAP (HEAT_EVAP)

Latent heat of evaporation for the normal boiling point temperature of the liquid. Only available if "show advanced" is checked. If Use database is checked, or if HEAT_EVAP is not specified, or if HEAT_EVAP is set to a negative value or zero, then HEAT_EVAP is calculated using properties from the FLACSCFD database for the pool liquid. When Use database is checked, then HEAT_EVAP is recalculated at each time step to account for the changing pool temperature.

3.7.15.24 KIN_VISC_A (KIN_VISC_A)

Kinematic viscosity for air. Only available if "show advanced" is checked. If Use database is checked, or if KIN_VISC_A is not specified, or if it is set to a negative value or zero, then KIN_VISC_A is calculated using properties from the FLACSCFD database for the air.

3.7.15.25 MOL_WEIGHT (MOL_WEIGHT)

Molecular weight of the pool liquid. Only available if "show advanced" is checked. If Use database is checked, or if MOL_WEIGHT is not specified, then MOL_WEIGHT is read from the FLACSCFD database.

3.7.15.26 PRANDTL_A (PRANDTL_A)

Prandtl number for air. Only available if "show advanced" is checked. If not specified, or not set to a value greater than zero, then a default value of 0.786 is used.

3.7.15.27 RAD_T (RAD_T)

This was used previously to define the initial radius of the pool but is no longer used and so is ignored by the simulator. Only available if "show advanced" is checked.

3.7.15.28 T_BOIL (T_BOIL)

This was used previously to set the boiling temperature for the pool liquid. The boiling temperature for the pool liquid is now calculated from the Antoine vapor-pressure equation, using properties for the pool liquid that are read from the FLACSCFD database. This parameter is therefore no longer used in the simulator. Only available if "show advanced" is checked.
3.7.15.29  **T_POOL (T_POOL)**

Initial temperature of the pool. Only available if "show advanced" is checked. If T_POOL is not specified, or is set to a value greater than the boiling temperature of the liquid, or is not greater than zero, then this is set to the boiling temperature of the pool liquid.

3.7.15.30  **Use radiation (POOL_USE_RADIATION)**

Only available in CASD for Fire scenarios. This specifies which of four different settings is used for a coupled pool-fire scenario, and is written to the scenario file as POOL_USE_RADIATION:

- 0 : use zero radiation from the flame
- 1 : use a fixed radiation level
- 2 : use “ambient” radiation (for backward compatibility), i.e., radiation calculated from the ambient temperature
- 3 : use the radiation field from the radiation model.

3.7.15.31  **Fixed Radiation (POOL_FIXED_RADIATION)**

Only available in CASD for Fire scenarios. This parameter can be used to control the level of radiative heat transferred to the pool surface, and is saved in the scenario file as POOL_FIXED_RADIATION. This parameter is used when Use radiation (POOL_USE_RADIATION) = 1, or when Use radiation (POOL_USE_RADIATION) = 3 and no radiation model is selected. Otherwise it is not used. The default value is 10000 W m\(^{-2}\).

3.7.15.32  **Minimum radiation (POOL_MINIMUM_RADIATION)**

Only available in CASD for Fire scenarios. This parameter can be used to control the minimum level of radiative heat that reaches the pool surface, and is saved in the scenario file as POOL_MINIMUM_RADIATION. This parameter is used when Use radiation (POOL_USE_RADIATION) = 1, or when Use radiation (POOL_USE_RADIATION) = 3, otherwise it is not used. The default value is 10000 W m\(^{-2}\).

3.7.15.33  **View factor Six-Flux (POOL_VIEW_FACTOR_6FLUX)**

Only available in CASD for Fire scenarios. This is required by the Six-flux radiation model that was implemented in earlier version of FLACS, however it is not used in FLACSCFD.

3.7.15.34  **Use turbulence (POOL_USE_TURBULENCE)**

Only available in CASD for Fire scenarios. This parameter determines how the turbulence above the pool surface is calculated, and is saved in the scenario file as POOL_USE_TURBULENCE:

- 0 : do not apply any turbulence modifications
- 1 : use \( u_{\text{ref}} = \text{velocity of vapor entering from the pool}, \text{set TKE} = 1.5\times(RTI\times u_{\text{ref}})^2 \)
- 2 : use \( u_{\text{ref}} = \text{velocity of vapor entering from the pool}, \text{set TKE} = \max(TKE, 1.5\times(RTI\times u_{\text{ref}})^2) \)
- 3 : use \( u_{\text{ref}} = \text{velocity of bulk flow just above the pool}, \text{set TKE} = 1.5\times(RTI\times u_{\text{ref}})^2 \)
- 4 : use \( u_{\text{ref}} = \text{velocity of bulk flow just above the pool}, \text{set TKE} = \max(TKE, 1.5\times(RTI\times u_{\text{ref}})^2) \).

Where RTI is the relative turbulence intensity, TKE is the turbulent kinetic energy and \( u_{\text{ref}} \) is the reference air velocity above the pool surface.
3.7 Scenario menu

3.7.15.35 Relative turbulence intensity (POOL_RT1)

The relative turbulence intensity above the pool, which is saved to the scenario file as POOL_RT1. Only available in CASD for Fire scenarios. If POOL_RT1 is greater than zero, then it is used to calculate the turbulent kinetic energy immediately above the pool surface.

3.7.15.36 Turulence length scale (POOL_TLS)

The turbulence length scale above the pool, which is saved to the scenario file as POOL_TLS. Only available in CASD for Fire scenarios. If POOL_TLS is greater than zero, then it is used as the turbulence length scale immediately above the pool surface.

3.7.16 Leaks

This section of the scenario menu allows the scenario parameters that represent leaks to be entered or updated. These values are written to the scenario file and to the leak file. For information about how leaks are represented in FLACS-CFD, see the Leaks section in the Technical reference chapter.

Attention:

CASD does not yet support all settings that may be present in a scenario or leak file (such as FAN, reverse, see below). When an existing scenario with these settings is opened in CASD, then these settings are not read correctly and are deleted when the scenario is re-saved. This only applies when the existing scenario uses a leak file, not when only non-time-dependent leak(s) are prescribed. See also the Leak file section in Input files for FLACS-CFD simulations. CASD may also overwrite leak files that are write-protected at file system level. It is recommended to always review the cl-file before running the Flacs simulator.

Remember to set the ER0 parameter under GAS_COMPOSITION_AND_VOLUME to a large number if the leak is a pure fuel leak.

3.7.16.1 Insert

This command is used to insert a new leak, the parameters for this leak are explained in the following pages. The maximum number of leaks in a scenario is 50.

3.7.16.2 Type

The type of leak must be specified as one of the following:

3.7.16.2.1 DIFFUSE A DIFFUSE leak is a no-momentum (the leaking gas has the same velocity as the surrounding flow) release of gas into the grid cell chosen. One must give a direction, but this direction will not be applied. For diffuse leaks, the leak control volume sides are not modified, keeping the original porosities.

3.7.16.2.2 JET A JET leak specifies a (low or high momentum) point leak or area leak. For jet leaks, the leak control volume will be closed on all sides initially by setting its area porosities to zero, and then the faces in the jet direction are opened up as specified by the OPEN_SIDES setting.

3.7.16.2.3 ENTRAINING Under specific conditions, the entrainment of air into a jet from a leak can be captured using empirical expressions and does not need to be explicitly calculated. This can save considerable computational cost, but is only appropriate when the following conditions are met:
• The leak is not in a highly congested area, i.e., the area between the leak and any obstacles is at least 22 times the leak diameter.

• The leak velocity is greater than 100 m/s.

• The jet flow-direction is aligned with a grid axis.

• The wind speed across the jet is less than 8 m/s at the leak, and for a distance of 22 leak diameters from the leak location, and is temporally constant.

• The air through which the jet flows for a distance of 22 leak diameters from the leak is quiescent.

• The diameter of the expanded leak area is less than 0.65 times the ambient grid cell size.

• Near-field concentrations are not of interest (using this setting causes the jet to be represented with coarser resolution close to the leak, and reported concentrations in the near-field are therefore more dilute).

For more information on this setting and recommendations for the grid, see Entrainment functionality. To activate this efficiency measure and avoid explicit entrainment calculations, the leak type should be set as Entraining, and slightly different constraints should be considered when refining the grid to capture the leak.

3.7.16.2.4 AIR An AIR leak will leak air (ER9) and is otherwise similar to a JET leak.

3.7.16.2.5 SUCTION A SUCTION leak is a negative point source, that is, gas/air will be removed at the leak location. See section Fans and suctions for details.

3.7.16.2.6 FAN A FAN leak does not add or remove mass, but assigns a fixed momentum to the gas in the concerned control volume. Fans must be aligned with one of the axis directions. A FAN leak can be located on a solid plate (of zero thickness) if the leak direction is normal to the plate; in this case, the flacs simulator will automatically open up the control volume corresponding to the position of the FAN. If a FAN with direction +X is specified, this will be translated into the string "!J+X=X:fan" in the leak file. See section Fans and suctions for details.

3.7.16.2.7 INERT A leak of INERT type leaks the specified inert gas rather than fuel. This requires Flacs inert to simulate.

3.7.16.3 Leak build-up

Leaks are built up in a time equal to 1% of the leak duration with a maximum of 1s. Shut-down is also gradual over 1s. These values can be changed manually in the leak file.

3.7.16.4 Leak concentration between ER0 and ER9

If a leak does not yield pure fuel, a concentration between ER0 and ER9 can be specified by manually changing the leak control string. Example: To get a concentration of 60% ER0 (mass fraction of ER0 is 0.6) and 40% ER9 (mass fraction of ER9 is 0.4) one must change leak control string in cl-file.

From:

"!J+X"

to:

"J+X:mix 0.6"
3.7 Scenario menu

3.7.16.5 Position

This menu is used to specify the leak position (in metres). For modelled-entrainment leaks, this is the initial position, which is redefined by the entrainment functionality.

3.7.16.6 Open sides

The **OPEN_SIDES** setting indicates which side(s) of the control volume affected by the leak should be opened up as the leak starts. **OPEN_SIDES** is a text string containing characters taken from "+−XYZ". A leakage directed in the positive x-direction should have **OPEN_SIDES** = "+X". Several sides may be open for a single leak, e.g. **OPEN_SIDES** = "+X−X" or **OPEN_SIDES** = "+XY−ZX". This will give two or more identical leaks in that the given leak rate will be applied for each open side. Thus a JET leak with **OPEN_SIDES** 'X−X' will have twice the rate of a leak with **OPEN_SIDES** 'X'.

A JET or AIR leak will generate flow in the directions specified by **OPEN_SIDES**, i.e., out of the control volume (mass source). A SUCTION leak will generate flow in the opposite direction, i.e., into the leak CV (mass sink).

When a FAN leak is considered, having downstream flow velocity pointing, e.g., in the positive z-direction, **OPEN_SIDES** can be set to "+Z" (and similarly for other axis directions).

The value of **OPEN_SIDES** is not relevant (i.e., ignored) when the direction cosines are given non-default values, i.e., other than [0.0, 0.0, 0.0]. For example, if the direction cosines are set [1.0, 1.0, 0.0] for a certain jet release, then it does not make any difference to this jet simulation whether **OPEN_SIDES** equals "+XY" or "+X" or any other value.

Note that in the cl-file the leak control string will contain the information regarding the kind of leak and **OPEN_SIDES** in short form, e.g. 'J+X' for a JET leak, 'J+X:lean' for an AIR leak, and 'J+X:reverse' for a SUCTION leak.

3.7.16.7 Start time

This is the time (in seconds) when the leakage should start. Normally one would allow a certain time for wind build-up prior to starting any leaks.

When a FAN leak is considered, **START_TIME** is set to the time when the FAN starts (typically this will be at time equal zero).

3.7.16.8 Duration

This is the duration (in seconds) of the leakage. The finish time for the leakage is the start time plus the duration. Hence, set TMAX (Maximum time) accordingly. The build-up time of the leak is 1 % of leak duration (maximum 1 second).

When a FAN leak is considered, **DURATION** can be set to a very large value if the fan is supposed to be active for the duration of the simulation.

3.7.16.9 The Outlet menu

The outlet menu can be expanded to show the following fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>AREA</td>
<td>0.1</td>
</tr>
<tr>
<td>MASS_FLOW</td>
<td>0.0</td>
</tr>
<tr>
<td>VELOCITY</td>
<td>10.0</td>
</tr>
<tr>
<td>RELATIVE_TURBULENCE_INTENSITY</td>
<td>0.15</td>
</tr>
<tr>
<td>TURBULENCE_LENGTH_SCALE</td>
<td>0.01</td>
</tr>
<tr>
<td>TEMPERATURE</td>
<td>6.0</td>
</tr>
<tr>
<td>DIRECTION_COSINES</td>
<td>0.0, 0.0, 0.0</td>
</tr>
</tbody>
</table>

If the **Leak Wizard** is used then these fields are automatically populated. Otherwise, the jet utility program can be used to calculate the appropriate values, or the **Steps when the jet utility program is not employed** can be followed.
3.7.16.10  Steps when the jet utility program is not employed

When the jet utility program is not employed, the user must:

1. Specify mass flow rate [kg/s] for the fuel jet leak (for release of the considered fuel composition).

2. Specify the velocity [m/s] of the jet after expansion to the ambient pressure. When the jet utility program is used with the single planar shock model, e.g., for a natural gas leak, the velocity [m/s] of the jet leak (after expansion to ambient pressure) is typically around 250 [m/s]. When the fuel is hydrogen, then this velocity [m/s] is typically around 600 [m/s]. Note that the speed of sound is quite different in methane (natural gas) and hydrogen. For example, at 0 Celsius and standard atmospheric pressure, the speed of sound is approximately 466 [m/s] in methane, and 1290 [m/s] in hydrogen.

3. Specify an assumed or estimated temperature [K] for the jet leak after expansion to the ambient pressure.

4. Use the fuel composition and the mass rate [kg/s], velocity [m/s] and temperature [K] from steps 1-3 to calculate the density of the fuel [kg/m³] after expansion, and the expanded leak area [m²].

The values specified or manually calculated in steps 1-4 are necessary for defining the outlet parameters that are required for simulation of the leak: area, mass flow, velocity, and temperature (cf. below).

3.7.16.11  Outlet: area

The effective cross-section area \( A \) of the leak outlet must be specified for any type of leak. For a specified mass flow \( \dot{m} \) (or volume flow \( \dot{V} \) and density \( \rho \)) the velocity \( u \) at the outlet should not be above the speed of sound. The following relations apply for the conditions at the outlet:

\[
\dot{m} = \rho \dot{V} = \rho u A
\]  \hspace{1cm} (3.10)

Assuming ideal gas properties:

\[
\rho V = n \mathcal{R} T = \frac{m}{M} \mathcal{R} T = mRT
\]  \hspace{1cm} (3.11)

where \( \mathcal{R} \) is the ideal gas constant (8314.3 [J/kmol·K]), \( T [K] \) the and the specific gas constant \( R \) is equal to the ideal gas constant divided by the molecular weight \( M \), the gas density at the outlet is:

\[
\rho = \frac{m}{RT} = \frac{\dot{m}}{\dot{V}}
\]  \hspace{1cm} (3.12)

\( \dot{m} = \rho u A \) Mass flow at the leak outlet (kg/m³/s)

\( \rho = \frac{p}{(RT)} \) Gas density at the outlet (kg/m³)

\( p = p_{amb} \) Pressure at the outlet is the ambient pressure (Pa)

\( R = \mathcal{R}/W \) Specific gas constant (J/kg/K)

\( W \) Mole weight for the given gas composition (kg/kmol)

\( T \) Temperature at the outlet (K) (0.0°C = 273.15K)

\( u \) Velocity at the outlet (m/s)

The relations shown above may be used to ensure that the outlet conditions are set in a consistent way. When a FAN leak is considered, AREA can in general be set to a value slightly less than the area of the CV face (e.g. 0.9 times the area of the Control Volume face, corresponding to 0.9 m² if you use 1m grid).

3.7.16.12  Outlet: mass flow

Specify the mass flow rate at the leak outlet here. Note that the mass flow takes priority if both mass flow and velocity are specified (see below). Make sure that the generated velocity is not larger than the speed of sound (approximately 340 m/s for air). Setting the outlet area large enough for a given mass flow rate causes the outlet velocity to be as low as desired. FLACS-CFD will report the generated velocity on the log file (rt-file). The mass flow rate must have a positive value (zero if not specified).
3.7 Scenario menu

3.7.16.13 Outlet: velocity

Instead of specifying a mass flow rate at the outlet (see above) you may specify the velocity. FLACS-CFD will report the generated mass flow rate on the log file (rt-file). If you specify both MASS_FLOW and VELOCITY then the mass flow rate takes precedence. The velocity must have a positive value (zero if not specified). The specified velocity should be lower than the speed of sound.

3.7.16.14 Outlet: relative turbulence intensity

This parameter must always be specified for a leakage. Normally a value in the range 0.01 to 0.10 will be appropriate. If the relative turbulence intensity cannot be obtained from any source (literature or experiments) for the given leakage, the following rough classification may be used:

- **0.01-0.03** Low turbulence intensity
- **0.03-0.06** Medium turbulence intensity
- **0.06-0.10** High turbulence intensity

Turbulence generated in the jet zone is generally of less interest than turbulence generated by the induced downstream flow field, in which case the $k-\varepsilon$ model in FLACS-CFD represents the actual modelling. It should be noted here that on the coarse grids normally used for dispersion calculations in large geometries the generated turbulence due to fluid stresses may be underestimated, and since numerical diffusion may be high on such coarse grids one might expect that the turbulent (or effective) mixing process is not represented with high accuracy.

3.7.16.15 Outlet: turbulence length scale

This parameter must always be specified for a leakage. It may be difficult to obtain the value for the turbulence length scale. It is generally related to the size of the nozzle or the feeding pipe for the leakage. A rough estimate is that the turbulence length scale is in the range of 10% to 20% of the nozzle diameter. When the turbulence length scale is small the dissipation rate of the turbulent kinetic energy is large, thus the turbulence will dissipate quickly for small diameter nozzles. In such cases, the turbulence generated by the downstream flow field is the important factor, and the $k-\varepsilon$ model in FLACS-CFD takes care of that.

3.7.16.16 Outlet: temperature

This parameter must always be specified for a leakage. For a leak of type FAN the downstream temperature is automatically inherited from the upstream temperature. The TEMPERATURE for a FAN leak can for example be set to the ambient temperature (but in general this is not necessarily equal to the upstream temperature).

3.7.16.17 Outlet: direction cosines

This is a vector which may be used to define the direction of an oblique jet. The direction cosines are the cosines of the angles between the direction vector and the three coordinate axes. See, e.g. [http://en.wikipedia.org/wiki/Direction_cosine](http://en.wikipedia.org/wiki/Direction_cosine) for further explanation, if needed. FLACS will normalise the vector, so the magnitude of the direction vector need not necessarily be equal to one (e.g. specifying [0.7071, 0.7071, 0.0] will have the same effect specifying [1.0, 1.0, 0.0]).

When the direction cosines are given non-default values (i.e., other than [0.0, 0.0, 0.0]), then the leak direction is entirely determined by the direction cosines and the OPEN SIDES setting is ignored. When an oblique jet release is modelled in FLACS-CFD, several CV faces are used to model the release. If two or more oblique jets have release points in the same CV or adjoining CVs (for the numerical grid used), it may happen that the total effective release rate [kg/s] for the jets is less than the sum of the release rates specified. User can check this based on the output in the simulation log file (rt*.dat3). If this happens,
consider refining the grid and/or moving the jet release location so that there are at least two empty CVs between any two leak cells.

See also:

Potential issues with oblique jet leaks.

Attention:

The specification of direction cosines for oblique jets is not compatible with modelled-entrainment leaks.

3.7.16.18 The Vessel menu

This menu was developed to calculate the outlet conditions for a leak from a pressurised vessel. Currently, it is not recommended to use the vessel menu. The jet utility program should be used instead.

3.7.16.19 Fans and suctions

**FAN** leaks are used to define fixed momentum leaks in a control volume (CV), e.g. to generate flow downstream of an exhaust fan. A **SUCTION** leak is a negative point source, i.e., it models the removal of fuel/air mixture. The syntax for defining such release types is similar to the case for defining a hydrocarbon release, i.e.,

**POSITION** the location of the centre of the fan

**SIZE** set to "0 0 0" in order to define a point source/sink

**OPEN_SIDES** the open side(s) of the leak control volume; for example, this would be +Z for **FAN** and **AIR** leaks for ventilation in upward (vertical) direction. For a **SUCTION** leak, having +Z as the open side, means that the gas being removed from the computational domain (due to suction) flows in the negative z-direction, namely into the +Z face of the CV.

**START_TIME** should presumably be set to zero or a low value (time when the fan starts)

**DURATION** should presumably be set to a very large value if the fans are supposed to be active for the duration of the simulation

3.7.16.19.1 Outlet menu

**AREA** can be set to a value slightly less than the area of the CV face (e.g. 0.9 times the area of the CV face; 0.9m² if you use 1m grid). The area of a single leak in FLACS-CFD must not be larger than the control volume face area where the leak is placed (unless an **area leak** is used).

**MASS_FLOW** self explanatory

**VELOCITY** not required if mass flow and area are specified

**RELATIVE_TURBULENCE_INTENSITY** suggest that 0.1 might be a suitable value

**TURBULENCE_LENGTH_SCALE** this would be 10% of the fan diameter

**TEMPERATURE** can for example be set to the ambient temperature. Actually; for a jet of type **FAN** the downstream temperature is automatically inherited from the upstream temperature (in general this is not necessarily exactly equal to the ambient temperature)

A **FAN** leak can be located on a solid plate (of zero thickness) where the leak direction is normal to the plate, since the flacs simulator will automatically open up whenever necessary the Control Volume (CV) corresponding to the position of the **FAN**.
3.7 Scenario menu

3.7.16.19.2 Leak file for FAN based on outlet parameters  
The core simulator flacs reads information from the leak file for the FAN (e.g. cl000000.n001 when the six digit job number is 000000 and leakage no. 1 is considered). This information is used during the simulation when modelling the FAN. If a valid leak file cl000000.n001 exists before the simulation starts, it will not be overwritten when the simulation starts unless the text string on the first line of the leak file begins with an exclamation mark ("!"). For example, if the file begins with 'J+X=X:fan', then the leak file will not be overwritten, but if the file begins with '!J+X=X:fan', then it will be overwritten. When there is no leak file before the simulation starts (or if any earlier leak file contains the exclamation mark and can thus be overwritten), then the leak file is created initially for the flacs simulation (before the actual physical modelling of the FAN), based on the outlet parameters. When the AREA \([m^2]\) and the VELOCITY \([m/s]\) are given positive values, and the MASSFLOW \([kg/s]\) is zero (default value), then the parameter 'AREA (m2)' in the leak file is given by the AREA from the outlet menu and it need not necessarily be exactly equal during gradual start up and gradual shut down. The parameter 'VEL (m/s)' in the leak file is given by the VELOCITY from the outlet menu. The parameter 'RATE (kg/s)' in the leak file is in general given the value (not necessarily exactly equal during gradual start up and gradual shut down of the FAN) of \(\dot{m}_{ER0} \ [kg/s]\), which is calculated by:

\[
\dot{m}_{ER0} = \rho_{ER0} U A
\]  

(3.13)

where \(U\) is the VELOCITY \([m/s]\) from the outlet menu, \(A\) is the AREA \([m^2]\) from the outlet menu, and \(\rho_{ER0} \ [kg/m^3]\) is the density of the fuel-air mixture for Equivalence Ratio equal ER0 at the ambient pressure and temperature equal to the TEMPERATURE \([^\degree Celsius]\) from the outlet menu. If, for example, ER0 equals zero and the TEMPERATURE from the outlet menu equals the ambient temperature (default value 20\(^\degree\)Celsius), then \(\rho_{ER0}\) is the density of pure air at ambient temperature and ambient pressure.

When the MASSFLOW \([kg/s]\) in the outlet menu is given a positive value, the parameter 'RATE (kg/s)' in the leak file is given the value of the MASSFLOW (again, not necessarily exactly equal to the specified value during gradual start up and gradual shut down of the FAN). Further, when the MASSFLOW \([kg/s]\) in the outlet menu is given a positive value, the value of the VELOCITY \([m/s]\) specified in the outlet menu, will not be used (you can just keep the default value 0.0 m/s for the VELOCITY in this case). The parameter 'VEL (m/s)' in the leak file will be given the value \(U\) \([m/s]\) (with possible deviations of \(U\) during gradual start up and shut down of the FAN leak), where \(U\) is calculated by:

\[
U = \frac{\dot{m}_{ER0}}{\rho_{ER0} A}
\]  

(3.14)

Here \(\dot{m}_{ER0} \ [kg/s]\) equals the MASSFLOW.

3.7.16.19.3 Physical modelling of FAN from parameters read from the leak file  
The core simulator flacs employs parameters read from the leak file when modelling the fluid flow through a FAN. The content of the leak file may, for example, look like this:

'!J+Y=Y:fan'

'TIME(s)' 'AREA(m2)' 'RATE(kg/s)' 'VEL(m/s)' 'RTI(-)' 'TLS (m)' 'T (K)'
0.0000E+00 5.0000E-01 1.1836E-01 2.0000E-01 5.0000E-02 8.0000E-02 2.9315E+02
1.0000E+00 5.0000E-01 1.1836E+01 2.0000E+01 5.0000E-02 8.0000E-02 2.9315E+02
8.9999E+02 5.0000E-01 1.1836E+01 2.0000E+01 5.0000E-02 8.0000E-02 2.9315E+02
9.0099E+02 5.0000E-03 1.1836E-01 2.0000E+01 5.0000E-02 8.0000E-02 2.9315E+02

In the above example the core simulator flacs will, after an initial gradual start up, fix the fan velocity \(U_{FAN}\) at the FAN leakage position to the value 20.0 m/s (cf. column 'VEL (m/s)' above). The area porosity at the FAN outflow position is then dynamically adjusted (whenever possible for the numerical grid chosen) so that the effective area \(A_{effective} \ [m^2]\) obeys:

\[
\dot{m}_{FAN} = \rho_{FAN} U_{FAN} A_{effective}
\]  

(3.15)

where \(\dot{m}_{FAN}\) equals (in this example) 11.836 kg/s (cf. column 'RATE (kg/s)' above), and \(\rho_{FAN} \ [kg/m^3]\) is the density of the gas at the position of the FAN (linear interpolation based on the upstream value and the
downstream value relative the position of the FAN). Note that \( \rho_{\text{FAN}} \) and \( A_{\text{effective}} \) will in general vary as function of time (depending on the fluid flow for the scenario considered), and the value of \( A_{\text{effective}} \) need not be equal to the corresponding value in the column 'AREA (m²)' in the leak file (for the example leak file above, \( A_{\text{effective}} \) is not necessarily equal to 0.5 m²). Note also that the time-dependent temperature values at the FAN position do not necessarily correspond to the temperature values listed in the column 'T (K)' above.

In some simulation cases, the value of \( A_{\text{effective}} \) calculated by the above equation may be larger than the area of the Control Volume (CV) face at the FAN outlet (for the numerical grid resolution employed during the simulation). This could for example happen when the specified fuel is a heavy gas like propane (heavier than air). Since the FAN model does not allow an effective area larger than the area of the CV face, you will then get a warning in the log file r000000.dat3 (if the job no. is 000000) regarding a LEAK EXCESS AREA; see also the section about Leak excess area. A solution can be to decrease the value of the AREA [m²] in the outlet menu (you will then get a higher value of \( U_{\text{FAN}} \) [m/s] when the MASS FLOW [kg/s] in the outlet menu is given a positive value, or a lower value of \( n_{\text{FAN}} \) [kg/s] when the MASS FLOW [kg/s] in the outlet menu is zero), and rerun the simulation (assuming that any earlier leak file contains the exclamation mark and can be overwritten). An alternative approach could be to remove any exclamation mark from an earlier created leak file for the FAN (i.e., replace the string 'J+X=X:fan' with the string 'J+X=X:fan') and manually edit the parameter values of this leak file before rerunning the simulation.

It is in general recommended that simulations are run until the flow behaviour has stabilised. This can, for example, be checked by employing an r3file command like:

```
run r3file r3123456.dat3 name=AVERAGES
```

See also:

Check also the guidance regarding air change rate.

3.7.16.19.4 Volume flux due to SUCTION balanced by volume flux entering region

Assuming uniform (ambient) pressure and temperature for ideal gas within the computational domain of the flacs simulation, the volume flux [m³/s] of gas entering the domain should, in order to keep constant pressure and temperature within the domain over time, be the same as the volume flux [m³/s] of gas leaving the domain (say corresponding to a SUCTION). This follows from the ideal gas law \( pV = NkT \) where \( p \) [Pa] is pressure, \( V \) [m³] is volume, \( N \) [-] is number of molecules, \( k \) [J/K] is the Boltzmann constant, and \( T \) [K] is temperature. Note that the total mass of gas within the domain may change over time even though the volume flux [m³/s] of gas entering the domain equals the volume flux [m³/s] of gas leaving the domain.

To see this, consider for example that a volume of 1.0 [m³] pure air enters the domain (in a certain time interval) and a volume of 1.0 [m³] methane-air mixture leaves the domain (during the same time interval); then there is increased total mass inside the domain (since the methane-air mixture has less density [kg/m³] than pure air, for the same thermodynamic condition).

Typically for ventilation/dispersion studies, the user would like to model (more or less) constant pressure and temperature within the domain over time (not a problem that the total mass of gas within the domain may change over time), and it would be helpful with a feature (functionality) so that the user could specify the volume flux [m³/s] for say an AIR leak, and specify the same volume flux [m³/s] for SUCTION (to set a perfect volume flux balance for a non-uniform density field). However, such a feature is currently not implemented in flacs. Currently the flacs simulator reads the mass flow rate [kg/s] 'RATE(kg/s)' from the leak file (e.g. cl<job no.>_n001) for the AIR leak and the SUCTION (similar for FAN). And thus if you have the same value of 'RATE(kg/s)' in cl-files for both the AIR leak and the SUCTION, the volume flux [m³/s] of gas entering the domain will in general not be equal the volume flux [m³/s] of gas leaving the domain (e.g. when you have non-uniform fuel concentration for methane-air mixture inside the domain, and thereby non-uniform density field).

The FLACS-CFD user may consider (when simulating non-uniform density field):

1. Specify leakages for the AIR leak and the SUCTION, so that the mass flow rate [kg/s] 'RATE(kg/s)' stored in the leakage file (cl-file) for the AIR leak is exactly the same as the mass flow rate [kg/s] 'RATE(kg/s)' stored in the cl-file for the SUCTION.
2. Whenever needed, consider to combine the AIR leak and the SUCTION with a (small) opening (open boundary condition), trying to minimize the effect of unwanted changes in pressure and flow field. (A measure trying to reduce the effect of not having the same volume flux for the SUCTION compared to the volume flux for the AIR leak.)

3.7.16.19.5 Verifying the volume flux leaving the computational domain due to SUCTION To verify that the volume flux leaving the computational domain due to SUCTION is as intended, you can use the r3file utility program to obtain the volume flux entering and leaving a specified region as function of time (how many time steps depends on how often 3D field data are saved to the r3-file).

See also:
Section air change rate.

You should make sure that every suction/leakage and any yielding pressure relief panel for your simulation case is well within the interior of the domain. Also note that if you, for example, have a suction within the domain, the net volume flux of the region is not necessarily exactly equal the volume flux of the suction at all times during a transient flow, due to the compressibility of the gas.

3.7.16.20 Leak Wizard

The leak wizard in CASD can be used to add/edit leaks when the scenario type in the 'Scenario Settings' dialogue is set to Dispersion, Fire, Gas Explosion or Pool. The wizard can be launched from the Run Wizard->leak wizard button available in the Scenario Settings section or from context menu for any leak in the Leaks section (right-click on the leak). The Wizard GUI has Input and Output pages. The Input page takes the input fields and the Output page displays the resulting profile, table and output data.

3.7.16.21 Input

The first page is used to set up a new, or edit an existing, leak.
**Figure 3.36: Gas phase input fields**

<table>
<thead>
<tr>
<th><strong>Leak Properties:</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Phase</td>
<td>Gas</td>
</tr>
<tr>
<td>Type</td>
<td>Point</td>
</tr>
<tr>
<td>Position</td>
<td>0.0 m</td>
</tr>
<tr>
<td>Direction</td>
<td>+x</td>
</tr>
<tr>
<td>Manual entry</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Reservoir Properties:</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Composition</td>
<td>Air</td>
</tr>
<tr>
<td>Volume</td>
<td>12000.0 m$^3$</td>
</tr>
<tr>
<td>Pressure</td>
<td>300.0 barg</td>
</tr>
<tr>
<td>Temperature</td>
<td>15.0 °C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Release Properties:</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Release type</td>
<td>Hole Size</td>
</tr>
<tr>
<td>Nozzle diameter</td>
<td>0.05 m</td>
</tr>
<tr>
<td>Discharge coefficient</td>
<td>0.85</td>
</tr>
<tr>
<td>Pseudo-source model</td>
<td>Single planar shock (default)</td>
</tr>
<tr>
<td>Time dependency</td>
<td>Steady state</td>
</tr>
<tr>
<td>Start time</td>
<td>0.0 s</td>
</tr>
<tr>
<td>Duration</td>
<td>0.0 s</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Atmospheric Properties:</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Atmospheric temperature</td>
<td>15.0 °C</td>
</tr>
<tr>
<td>Atmospheric pressure</td>
<td>1.0 bara</td>
</tr>
</tbody>
</table>

[Image of the interface showing input fields for leak simulation in FLACS-CFD v22.2]
### Leak Properties:

1) **Phase:** Gas and liquid (flashing) are supported.
2) **Type:** Point and Area leaks are supported.
3) **Position and Direction:** Position of leak in world coordinates. Direction in +xyz/-xyz in any single direction or diffuse (only applicable in Gas phase) can be enabled for diffuse leak type.
4) **Manual Entry:** When manual entry is selected, the leak outlet parameters must be entered manually on the next page. This is useful when e.g. obtaining values manually or from other tools. In case of Area leak the calculated area is provided by wizard.

---

#### Figure 3.37: liquid (flashing) phase input fields

![Image of leak input fields](image)

---
Reservoir properties:

1) **Composition**: By default, the value for this field is taken from the volume fractions in the Gas composition and volume section of the scenario. This value can be changed manually for gas leaks. For releases of mixed gases, FLACS-CFD normalizes the volume fraction for each gas by dividing by the sum of all the specified volume fractions. For example, "methane=1, ethane=2" is interpreted as a mixture comprised of 2/3 ethane and 1/3 methane (by volume) and this is used to calculate kappa and the molecular weight for the mixture. For liquid-phase releases, only one species can be selected. For AIR AIR leaks, the value for this field should be AIR (this is the default value for gas leaks if no species is defined in the Gas composition and volume section of the scenario.)

Release properties:

1) **Nozzle Diameter**: Only available for point leaks. For area leaks, the area is calculated from the leak size.
2) **Release rate**: Only available for gas releases. For area leaks, the release rate is given precedence over the leak size in the calculations.
3) **Pseudo-source model**: single planar shock (default) or Ewan-Moodie (for a gas release, there is an ‘+air entraining’ option that can be combined with either of these models). Only the single planar shock model is supported for liquid (flashing) releases.

Attention:

For an entraining-type leak, the pseudo-source model should be selected with ‘+air entrainment’.

4) **Time dependency**: This can be steady, steady (limited) or time dependent for gas releases. Time-dependent calculations are not yet supported for liquid (flashing) releases.

Advanced options:
Advanced option enables the user to view and edit further options, such as the heat transfer coefficients, wall
temperature and relative turbulence intensity. The equation of state supports the Ideal gas law and the Real 
gas law. The Real gas law is only supported for gas releases that are 100% hydrogen.

![Advanced options](image)

**Figure 3.40: Advanced options**

**Area Leak Input:**

For area leaks there is an additional input page where the leak size, shape and profile should be entered.

1) **Size:** Both the sizes must be nonzero. The direction of the normal has to be specified as one coordinate 
direction (the extents are then assigned to the two other dimensions in order). In case of an area leak, the size 
specified at this stage is the actual surface of the leak, which may be smaller than the area of the leak box 
(e.g. if the leak is elliptic).

2) **Shape:** Here the leak shape (Elliptical or Rectangular) is specified.

3) **Profile:** Gaussian, Uniform or Parabolic profiles are supported where Gaussian is available only for 
Elliptical shape.

![Area leak inputs](image)

**Figure 3.41: Area leak inputs**

### 3.7.16.22 Output

On the output page, a summary of the compiled information about the leak is shown. The information about 
the leak is added to the scenario file and to the leak file.
In case of time-dependent gas-phase leaks, the wizard shows the leak profile first then the summary and leak info.
3.7 Scenario menu

3.7.17 Automatic grid refinement for jet leaks and jet fires

There is a plugin available that may help to refine the grid around leaks for point leaks, following the recommended configuration for the grid. It is based on the refine function in gm. The refinement plugin is available by right clicking on the leak you want to refine around, in the scenario menu leak section.

Figure 3.45: Refine grid plugin.
The plugin sets the refined cell sizes (in all dimensions) to

\[ CellSize_{new} = \sqrt{\text{refinement factor} \cdot \text{expanded leak area}} \]  

(3.16)

The refinement factor is shown to right of the **Adjust Refine Factor** checkbox and can be entered manually if this box is checked. By default, the refinement factor is 1.25 for standard **point leaks** and is 9 for **modelled-entrainment leaks**.

The refinement region (i.e., which cells in the grid are set to the new cell size) includes the cell(s) that contain the leak, plus one cell either side of this in the plane of the leak. The refinement region extends in front of the leak by three cells. Large size differences between neighbouring cells are not recommended in FLACS-CFD, so grid cells beyond the refinement region are adjusted to ensure a smooth transition between the refined cell size and the size of cells in the background grid, using the same smoothing as is implemented by the **Smooth** tool in CASD. The total extent of the region of affected cells (the refinement region plus cells affected by the smoothing) is shown in the wizard as the **Grid region affected**. Clicking the **Apply** button updates the Geometry view to show the new grid. Clicking **OK** keeps this grid, and **Cancel** brings back the original grid.

**Note:**

For a fire scenario with a jet leak, the plugin first estimates the flame shape using the calculations in the **Jet flame shape** section of the **Technical Reference** chapter. The estimated shape is used to create a core domain, following the **recommendations for configuring the grid**. Cells in this core domain are then refined around leak as above.
3.7 Scenario menu

3.7.18 Automatic grid refinement for pool fires

The grid guidelines require an estimate of the flame shape to setup the core domain for pool fire simulations simulation. The flame shape is used to refine the current grid so that it follows the guidelines related to the core domain. This dialog is opened by right clicking in the "Pool" section of the "Scenario menu" to open the context menu, and select "Refine grid". A grid that fulfills the requirements for the total domain should be set up before using the "Refine grid for pool fire" dialog, e.g. by using the Quick Grid functionality.

Figure 3.46: Example of grid before (top) and after (bottom) grid refinement around leak.
When opening the "Refine grid for pool fire" the flame shape for the current pool setup is estimated using the calculations in Maximum pool area and flame shape for pool scenarios, assuming a fixed burning rate of 0.15 kg/m²s, no wind effect, and no bund height. Based on the resulting flame shape, a "near pool region" and a "flame region" is calculated and used as the basis for the grid refinement. The changes to the current grid will immediately be reflected in the geometry window, and the grid will be permanently changed by closing the dialog by clicking the "Ok" button.

The "Z position of pool" is initiated with the current pool position, but as the final position of a pool is the result of the pool being "dropped" onto the geometry, this value should be manually adjusted so that the "near pool region" is placed at the correct position.

The start- and end-position of the regions along the individual axes of the coordinate system can be manually adjusted by writing directly to the input boxes or using the up and down arrows at the right side within the boxes. If the "Z position of pool" value is changed, the start and end values for the two regions will be overwritten by the algorithm that computes the flame shape.

Comparisons between the original grid and the refined grid are available at the bottom of the dialog, and show the effect of the grid refinement algorithm.

**Note:**

The cell size in the core domain is limited to maximum 1 meter, and the algorithm is not implemented for an unconfined pool.

### 3.7.19 Ignition

In case of a gas/dust explosion or fire simulation, specify the location and size of the ignition source. In addition, give a time and duration for the ignition. The available parameters here are as follows:

- **POSITION_OF_IGNITION_REGION** 0.0, 0.0, 0.0 (m)
- **DIMENSION_OF_IGNITION_REGION** 0.0, 0.0, 0.0 (m)
- **TIME_OF_IGNITION** 0.0 (s)
- **DURATION_OF_IGNITION** 0.0 (s)
- **RADMAX** 99999.0 (m)
The ignition region can be a point (0D), a line (1D), a plane (2D) or a volume (3D). The `DURATION_OF_IGNITION` parameter makes it easier to ignite fire scenarios at the start of the simulation. A failed ignition is reported only after the specified ignition duration is elapsed. The `RADMAX` parameter will be ignored by all FLACS-CFD versions after 1998; it is only present for backward compatibility.

### 3.7.19.1 Gas/dust explosion ignition

If the ignition point is inside a partially blocked control volume, the flame may quench. It is therefore not recommended to ignite inside a partially blocked control volume. But if you choose to do so, and if there are any problems obtaining a proper ignition and flame propagation, try to increase the `DIMENSION_OF_IGNITION_REGION` side lengths up to about 0.05 to 0.10 m.

When modelling gas explosions, the ignition in FLACS-CFD is usually set to occur in just one control volume. The ignition region should normally be a point (or 0D), as this represents the “spark ignition” generally seen in practical situations. However, larger ignition regions such as 1D or 2D can be used when doing benchmarking against experiments, including representing stronger ignition with multiple ignition locations. The time and duration of the ignition may also be specified. This is useful for igniting gas clouds that are generated during gas dispersion simulations. In a normal gas explosion simulation, the time of ignition should be set to zero.

**Attention:**

Avoid locating the ignition point exactly on a grid line, instead position it at the centre of a control volume. Usually, it is best to specify the ignition point according to the grid, and not according to the geometry. For example, walls are sometimes adjusted to the nearest grid line in the porosity calculations but the ignition location is not changed. This could result in an ignition location that is initially on one side of a wall becoming a location on the other side of the wall, causing the simulation to calculate ignition on the ‘wrong’ side of the wall!

![Figure 3.48: How to position the ignition](FLACS-CFD_v22.2_User's Manual)
Attention:

For explosion simulations it is not recommended to use big 3D ignition regions since this may lead to more violent explosions with overpredicted explosion overpressures.

3.7.19.2 Fire Ignition

When modelling jet- or pool fire scenarios, the flame duration is usually long compared to the initial flash fire or deflagration. Therefore any inaccuracies in the heat radiation and smoke results in the initial flash fire phase – introduced, for example, by using a larger ignition region – will be negligible compared to the relatively long jet- or pool fire phase. In most cases it is therefore recommended to use a large 3D ignition region to ignite jet and pool fires. This ensures that the jet or pool is ignited, without having to rerun the simulation to determine a flammable region for ignition. A typical large 3D ignition domain will be somewhere between 5m-20m in all directions relative to leak or in the vertical direction above a surface area leak. In the log file the FLACS-Fire solver will only report the cells where ignition was successful, to keep it concise.

In a jet or pool fire simulation, the time of ignition should be set so that it occurs with a short delay after the leak has started. This allows some flammable gas to accumulate, to make ignition more likely. A typical time would be 0.1s - 0.25s after a jet leak has started or 0.5s-2s after the start of a diffuse leak.

For simulations where the exact ignition location needs to be modelled (e.g. combination of deflagration and jet-fire or incident investigation), it is recommended to run the dispersion without ignition first, with a monitor point in the ignition location monitoring fuel concentration. The simulation can then be rerun with the ignition time set within the range that there was a flammable mixture at the ignition point.

As mentioned above, the DURATION_OF_IGNITION parameter is highly recommended for fire simulations. For the steady state cases this will help to minimise the peak in output variable immediately after ignition due to delay in ignition.

Note:

It is not yet possible to define multiple ignition sources or force ignition of a fire.

3.7.20 Water spray

Water deluge systems can be an effective way to mitigate the consequences of gas explosions in several situations. The mitigating effect has been seen in many experiments. However, the phenomena related to the interaction between water sprays and an accelerated flow field and the flame are quite complex. The explosion accelerates the flow as well as the water droplets. The acceleration depends on the local flow conditions, and the droplet size. When hydrodynamic forces become large enough to overcome surface tension, the droplets break up (again dependent on the droplet size). When the droplets are small enough (either because the droplets produced by the nozzle are small, or because large droplets break up into small droplets when the flow is accelerated), they tend to reduce the burning rate due to cooling of the flame, and dilution of the gas mixture by evaporation. However, large droplets in the flame region (before break-up of the large droplets) tend to increase the turbulence level and thereby increase the burning rate. So there are competing effects when using a water spray system. In some cases water spray may even increase the maximum overpressure of an explosion.

The problem when modelling water sprays is how to represent all this on a coarse simulation grid. It was necessary to do simplifications, and also to use experiments performed for tuning of the models. Two distinct effects were identified.

1. The acceleration of the flames due to turbulence from the sprays and the presence of the droplets.
2. The reduction of burning rate experienced in certain situations because of the water sprays.

For each of the nozzles an acceleration factor, denoted F1, is determined. F1 is used to increase the burning rate if any watersprays are present. A quenching factor, denoted F2, is also determined. F2 is used to reduce the burning rate if the conditions for droplet break-up are present.
3.7 Scenario menu

Remarks:

Due to model simplifications, there is no need for a very accurate positioning of the sprays. In regions where sprays of the same type will overlap, you should define one water spray region for the whole system. If more than one waterspray region is to be defined, make sure that they do not overlap, otherwise FLACS-CFD will stop. In FLACS-CFD, the droplets are assigned a velocity but the transportation of droplets is not modelled. Regions where it is obvious that a lot of droplets will be transported ahead of the flames (for instance directly outside vent openings), should be included in the spray region (i.e. larger water spray regions should be defined).

The models are validated in stoichiometric gas concentrations in a 180 $m^3$ vented box, with various obstruction levels, in a 50 $m^3$ model of an offshore module, and also in full-scale experiments. It is reasonable to believe that the models represent mechanisms in connection to water mitigation well.

The water spray model implemented in the FLACS-CFD code is relatively simple. One or more non-overlapping water spray regions are defined. In each region there is assumed to be droplets of a given diameter (before break up), and a given water volume-fraction. If the relative velocity between the droplets and the gas flow exceeds a so-called critical break-up velocity (depending on the diameter of the droplet), it is assumed that the droplets break up.

Two non-dimensional factors are employed in the numerical model. When there is water in the reactive mixture (i.e. inside a water spray region), this is assumed to enhance the burning rate. Before break-up of the droplets, the burning enhancement-factor denoted $F_1$ [-] (positive number) is multiplied by the laminar burning velocity and added to the ordinary burning velocity (i.e. the burning velocity employed in the FLACS-CFD code without any water spray, this burning velocity is in general in the turbulent regime) to give the effective burning velocity with water spray. When the droplet break-up criterion is fulfilled, the burning velocity (without break-up) is multiplied by the burning reduction-factor $F_2$ [-] (positive number less than 1), to give the effective burning velocity

$$S_{water} = (S_{turb} + F_1 \cdot S_{lam}) F_2$$  \hspace{1cm} (3.17)

The water spray model is validated for explosions with stoichiometric fuel-air mixtures. For non-stoichiometric fuel-air mixtures it is assumed that the pressure reduction caused by water spray might be slightly under predicted.

Below are the parameters which may be visible in the scenario file

```
| INSERT | 1 |
| POSITION | 0.0, 0.0, 0.0 [m] |
| SIZE | 1.0, 1.0, 1.0 [m] |
| VOLUME_FRACTION | 0.2 [litre/m^3] |
| MEAN_DROPLET_DIAMETER | 300.0 [micrometer] |
| NOZZLE_TYPE | *FACTORs: 10.0 0.3* [-] |
```

3.7.20.1 Insert

Integer identifying the water spray region considered. A maximum number of 25 regions may be defined in the current version of the FLACS-CFD code.

3.7.20.2 Position

Cartesian coordinates [m] of the corner of the box-shaped water spray region (the corner with lowest value of the coordinate in each axis direction). Make sure not to define any overlapping regions. The FLACS-CFD code will assign the nearest grid line to your given position as the actual position of the water spray region. In other words, FLACS-CFD will make the water spray region stick to the grid lines.

3.7.20.3 Size

The dimension [m] in each of the axis directions is given for the box-shaped water spray region. All three dimensions should be positive. Make sure not to define any overlapping regions, the FLACS-CFD code
will assign the nearest grid line to your given (position + size) as the actual (position + size) of the water spray region. In other words, FLACS-CFD will make the water spray region stick to the grid lines. The FLACS-CFD code will issue an error message and stop if you have defined any overlapping water spray regions. You should check the position and size parameters if this error occurs.

### 3.7.20.4 Volume fraction

The liquid water volume fraction $VOLUME\_FRACTION$ [litre/m$^3$] is defined by volume of liquid water in litre divided by total volume in cubic meter (one cubic meter equals 1000 litre), inside the water spray region. Thus, if the liquid water volume fraction is 0 [litre/m$^3$] there is no liquid water, and if it is 1000 [litre/m$^3$] there is only liquid water.

The liquid water volume fraction can be estimated using the $\beta_{water}$ formula found here. Typical values for the liquid water volume fraction for water spray applications, are expected to be in the range 0.1-0.4 [litre/m$^3$] (Dale, E. K., 2004).

If the parameter is less than 0.01 [litre/m$^3$] (expected to be very unlikely for industrial water deluge applications), the FLACS-CFD code will give you a warning that the water spray is assumed not effective in the numerical model. As long as the parameter is larger than the minimum value of 0.01 [litre/m$^3$], the water spray model in FLACS-CFD is effective.

In the current version of FLACS-CFD setting the parameter to any value above 0.01 [litre/m$^3$] will give the same numerical results in the simulation, but note that the value of the liquid water volume fraction is important when estimating the two factors $F1$ and $F2$ for the parameter NOZZLE\_TYPE (more details about $F1$ and $F2$ can be found here).

### 3.7.20.5 Mean droplet diameter

The mean diameter $MEAN\_DROPLET\_DIAMETER$ [micrometer] of the water droplets before break-up due to acceleration of the gas flow, is given by the user. In the water spray model it is assumed that all droplets have the same size, and that the droplets are uniformly distributed in space inside the water spray region. This is an approximation. In most real situations there will be a droplet size distribution, which in most cases is non-uniform in space, i.e. the distribution may change from one region to another region.

In the water spray model, the mean droplet diameter is defined to be the so-called Sauter diameter. This diameter is defined by the volume-based mean diameter cubed divided by the area-based mean diameter squared. The Sauter diameter depends on the operating water-pressure forcing water out the nozzle. The empirical relation

$$ D_{\mu m} = 10^3 P^{-0.333} $$

($D_{\mu m}$ in [micrometer] and $P$ in [barg]) seems to give good estimates for the Sauter diameter over a range of values for the water pressure, and for various nozzles. However, for some special nozzle types giving very large (e.g. nozzle type LDN) or very small (e.g. nozzle type P120) droplet diameters, this empirical relation may be less accurate. One may then try to estimate otherwise the Sauter diameter for the water-pressure considered. You may use either your estimate or data given by the vendor of the nozzle.

### 3.7.20.6 Nozzle type

The parameter NOZZLE\_TYPE should be set equal to the text string "FACTORS: F1 F2" (note the two occurrences of the double quote character " in the text string), where F1 and F2 are the numerical values of the two factors. The two non-dimensional factors $F1$ and $F2$ are modelled by

$$ F1 = 14U_z\beta_{water} $$

and

$$ F2 = \frac{0.03}{D_{\mu m}\beta_{water}} $$
3.7 Scenario menu

where $U_z$ [m/s] is the average droplet velocity vertically downward (absolute value), $\beta_{\text{water}}$ [per thousand] is the water volume-fraction, and $D_{mm}$ is the Sauter diameter measured in [mm] ($D_{mm} = 10^3 D_{mm}$). How to obtain the Sauter diameter (mean droplet diameter) is described above. If the nozzle spreads the water horizontally, the droplets will soon fall down with a constant velocity (gravity forces are balanced by drag forces). This constant velocity depends on the droplet diameter. It can be estimated from the empirical relation

$$U_z = 2.5(D_{mm})^{0.94}$$

(3.21)

where the units of $U_z$ is [m/s] and the units of $D_{mm}$ is [mm]. For some nozzle types the droplets leave the nozzle with a significant velocity component vertically downward. In this case the average downward velocity should be estimated otherwise (giving a larger value than from the expression above). The water volume-fraction is estimated by

$$\beta_{\text{water}} = \frac{n(Q/60)}{X_{\text{length}}Y_{\text{length}}U_z}$$

(3.22)

where $X_{\text{length}}$ [m] is the length in x-direction of the assumed rectangular waterspray region, $Y_{\text{length}}$[m] is the length in y-direction (it is assumed that the xy-plane is the horizontal plane), $n$ is the number of nozzles (it is assumed that all of the nozzles within the same waterspray region is of the same type and with the same water flow-rate), and $Q$ is the water flow-rate [litre/min] for a single nozzle (thus $Q/60$ is the water flow-rate in units of [litre/s]). The water flow-rate depends on the operating water-pressure. It is assumed that the flow-rate is related to the water pressure $P$ [barg] by

$$Q = k\sqrt{P}$$

(3.23)

where the so-called $k$-value of the nozzle depends on the type of nozzle considered.

Note:

In situations where there is only limited information about the nozzles, the factors $F1$ and $F2$ can be estimated using the simplified formulae: $F1 = 0.233 \cdot W\,A\,R$ and $F2 = 4.5/\,W\,A\,R$ where $W\,A\,R$ is the water application rate in litre/m² · min.

Attention:

The deluge model in FLACS-CFD only models the effect on deflagrations. It cannot model the effect of deluge on dispersion or fire scenarios.

3.7.21 Louvre panels

Warning:

The louvre panel model has not been thoroughly validated, and some limitations have been identified. Especially the thickness of the louvre slats (section Area porosity) does not seem to have a large enough effect on the results. As the louvre panel model is relatively complicated to set up, it is recommended to use a porous plate in most situations.

Louvre panels are common in offshore installations and also in land-based process industry. The louvre panels affect the flow field both due to drag forces, and deflection of the flow downstream of the panel, determined by the geometry of the louvre slats. A new sub-grid model for flow through louvres is described in detail in (Salvesen, 1996a). Some validation exercises using the new sub-grid model are documented in (Salvesen, 1996b).

A louvre panel is assumed to consist of slats mounted on a frame. The louvre slats may have an arbitrary form depending on the type of louvre panel considered, but it is assumed that the slats of a louvre panel are more or less equal in shape, uniformly oriented, and uniformly distributed. It is assumed that the velocity vector downstream of the louvre panel is forced to be within a fixed plane (a mathematical plane of zero...
thickness) determined by the louvre slats, when the flow exits the louvre, independently of the upstream velocity.

To be specific, let us consider an example: A louvre panel is oriented with normal in x-direction. The louvre slats define a plane with unit normal vector in the xy-plane, \((-\sin(q), \cos(q), 0)\), where \(q\) is an angle with absolute value less than 90°. The angle \(q\) is defined as the angle between the slats of the louvre (assuming that the slats are more or less plane, or if they are curved that they define a tangent at the exit on the downstream side) and the normal of the louvre (in our example the x-axis). The assumption is that the projection of the downstream velocity vector for flow in positive x-direction, on to the xy-plane, is directed a fixed angle \(q\) relative the positive x-axis (corresponding to the tangent vector \((\cos(q), \sin(q), 0)\)), independently of the upstream velocity. How good this assumption is in practice depends on several factors: the form (plane or curved) and the thickness of the slats, and how wide the slats are compared to the distance between two neighbouring slats. The component of the velocity vector in z-direction, is assumed to be unaffected by the presence of the louvre panel.

In the example above, a louvre panel is oriented in x-direction. It may also be oriented in y- or z-direction. In general the louvre slats define two different planes, one plane for flow exiting in positive direction and one plane for flow exiting in negative direction. Often these two planes are identical. This is the case e.g. if the slats are of rectangular shape. But if the slats are e.g. V-formed, the two planes are distinct. The two planes may be arbitrary oriented as long as two conditions are satisfied: Firstly, none of the planes should coincide with the louvre plane itself (this would imply that the louvre panel is completely blocked). In the numerical code it is checked that the angle between the unit normal of the louvre plane and the unit normal of the plane determined by the slats (for flow in positive or negative direction), is not too small, that is less than one degree. Secondly, if the plane determined by the slats for flow in positive direction is distinct from the plane for flow in negative direction, the vector cross product of the normal vectors of these two planes should lie in the louvre plane (in other words, these two normal vectors and the normal of the louvre plane all lie in the same plane).

The sub-grid model affects how the momentum flux is calculated for the face of the staggered control volume adjacent to the louvre plane on the downstream side. The velocity of a control volume is determined by a balance of momentum fluxes over all the faces of the control volume. The sub-grid model gives the momentum flux at the CV face adjacent to the louvre panel, the momentum fluxes at the other CV faces are unaffected by the presence of the louvre panel.

The sub-grid model it is assumed that the total drag force can be represented as the sum of three terms; drag due to acceleration of the magnitude of the velocity of the flow, drag due to bending of the flow, and drag due to friction. In the description of these three terms below, it is assumed that the flow is in positive direction (either x-, y- or z-direction). The subscript \(e\) (east) corresponds to downstream values, the subscript \(w\) (west) corresponds to upstream values. A similar description is valid for flow in negative direction (not given here). Drag (absolute value) due to acceleration of the flow is represented by:

\[
F_{D,acc} = C_{acc} \max \{ 0.05 \rho_e [U_e]^2 \rho_w [U_w]^2 \} \tag{3.24}
\]

in the case that the plane determined by the louvre slats for flow in positive direction is the same as the one for flow in negative direction. Here \(\rho\) is the density, and \(U\) is the velocity vector. The drag coefficient \(C_{acc}\) is typically set to 1. No contribution from drag due to acceleration is included if the norm of the velocity vector upstream is larger than the one downstream (this would correspond to a pressure increase instead of a pressure drop over the louvre).

In the case that the plane determined by the louvre slats for flow in positive direction is distinct from the one for flow in negative direction (e.g. if the slats are V-formed), it is assumed that the drag is represented by:

\[
F_{D,acc} = C_{acc} \left[ \max \left\{ \frac{0}{0.5 \left( \rho_{int} [U_{int}]^2 - \rho_w [U_w]^2 \right)} \right\} + \max \left\{ \frac{0}{0.5 \left( \rho_e [U_e]^2 - \rho_{int} [U_{int}]^2 \right)} \right\} \right] \tag{3.25}
\]

Here \(U_{int}\) is the so-called intermediate velocity vector in the plane determined by the louvre slats for flow in
negative direction. It is assumed that the velocity vector first is forced to be within this plane, and then is forced to be within the plane determined by the louvre slats for flow exiting in the positive direction.

Drag due to bending of the flow is modelled as:

$$ F_{D,bend} = 0.5 \rho_w [U_w]^2 C_{bend}(\alpha_1 + \alpha_2) $$  (3.26)

where $\alpha_1$ is the angle between the upstream and the intermediate velocity vector, $\alpha_2$ is the angle between the downstream and the intermediate velocity vector, $0 \leq \alpha_i < \pi, i = 1, 2$. The intermediate velocity vector and the downstream velocity vector are identical when the two planes determined by the louvre slats (for flow exiting in positive and in negative direction) are identical (and then $\alpha_2 = 0$). The coefficient $C_{bend}$ (units of $1/radian$) could be estimated from experiments or fine grid simulations. In the validation simulations a value of $C_{bend} = 0.11/(\pi/4)$ is used. This corresponds to a result found in the literature; a resistance coefficient for flow in pipes of 0.11 for a bending of $\pi/4$ radians. Confer (Salvesen, 1996) for more details.

If the flow is neither bent nor accelerated, there is still a drag due to skin friction assumed to be represented by

$$ F_{D,fric} = 0.5 \rho_w [U_w]^2 C_{fric} $$  (3.27)

where $C_{fric}$ is a drag coefficient.

The coefficients $C_{acc}$, $C_{bend}$, and $C_{fric}$ are expected to depend on the specific type of louvre considered, and on both the Reynolds number and the Mach number of the flow. The direction of the upstream velocity vector may also be of importance. To investigate how the coefficients depend on the various parameters mentioned above, seems to be a challenging task. Performing fine-grid simulations or experiments are ways of approaching the problem. In the present model it is assumed that the coefficients depend on only the geometry of the louvre (neglecting the dependence on the other parameters mentioned above).

If experimental values of the coefficients $C_{acc}$, $C_{bend}$, and $C_{fric}$ are known for the specific type of louvre panel considered, these values should be used in the numerical calculations. In many cases only the so-called pressure-loss coefficient (sometimes in the literature it is also called pressure-drop or resistance coefficient) is known from experiment. This coefficient is defined by

$$ p_w - p_e = 0.5 \rho_w [U_w]^2 C_{pressure} $$  (3.28)

for upstream flow in positive direction with velocity vector pointing normally on the louvre plane. In the case of flow with low Mach number (incompressible or nearly incompressible fluid), the static pressure drop over the louvre panel is essentially balanced by the drag force (i.e. the component of the force from the louvre on the fluid along the normal of the louvre plane) per unit area (see (Salvesen, 1996a) for details).

Thus for incompressible flow, when $U_w = U_e$, with upstream velocity vector pointing normally on the louvre plane, the pressure-loss coefficient can be related to the coefficients $C_{acc}$, $C_{bend}$, and $C_{fric}$ by

$$ C_{pressure} = C_{acc}(1/(\cos^2 \theta) - 1) + C_{bend}\alpha 1 + C_{fric} $$  (3.29)

where the relation $|U_e| \cos \theta = U_e$ is utilised (confer the example above). Here it is assumed that the plane determined by the louvre slats for flow in positive direction is the same as the plane for flow in negative direction. One approach is to set $C_{acc}$ equal 1 (this value is supported by theoretical considerations, cf. (Salvesen, 1996a)), set $C_{bend}$ equal zero or a guessed value based on experimental results found in the literature (e.g. set $C_{bend} = 0.11/(\pi/4)$ as described above), and set $C_{fric}$ so that the relation above is satisfied for given values of $C_{pressure}$, $C_{acc}$, and $C_{bend}$. Another approach is to set both $C_{acc}$ and $C_{bend}$ equal zero, and set $C_{fric}$ equal $C_{pressure}$. Both these approaches are expected to give good results when performing simulations. If not even the pressure-loss coefficient $C_{pressure}$ is known, one may use a guessed value for this coefficient.

Values of the pressure-loss coefficient are reported in the range from 9.8 to 13.9 for commercially available louvre panels for industrial use. So a value in the range, say between 10 and 11, is in many cases expected to be a reasonable estimate for the pressure-loss coefficient.

Up to 100 louvre panels can be defined. An example of a setup for a louvre panel is shown below:
### 3.7.21.1 Name

Character string identifying the louvre panel considered (not used by FLACS-CFD).

### 3.7.21.2 Position

Cartesian coordinates of the corner of the louvre panel (the corner with lowest value of the coordinate in each axis direction).

### 3.7.21.3 Size

The louvre panel is assumed to be of rectangular shape. The dimension in each of the axis directions is given. One dimension should be zero, this shows how the louvre panel is oriented, and defines the louvre plane. If e.g. the dimension in x-direction is zero, the normal of the louvre plane points in x-direction. The other two dimensions should be positive.

### 3.7.21.4 Material

The material to be used for modelling and visualising the louvre panel; see the section on panel materials for remarks on the usage.

### 3.7.21.5 Normal vector slats positive

Cartesian coordinates of normal vector of plane determined by the louvre slats (ribs) for flow in positive direction. This vector need not be specified as a unit vector. But the normal vector should not be parallel to the normal vector of the louvre plane. This would mean that the louvre panel is completely blocked by the louvre slats (if e.g. the louvre panel is oriented in x-direction, to specify the normal vector determined by the louvre slats as $(1, 0, 0, 0, 0)$ would be an illegal choice). The normal vector need not lie in the xy-, xz-, or yz-plane. To specify the vector as for example $(1, 0, 1, 0, 1)$ would be a valid choice.

If for example the louvre panel is oriented in x-direction, and the normal vector determined by the louvre slats is given by $(\sin(q), 0, \cos(q))$ (here q is an angle with absolute value less than 90°), the interpretation is the following: The angle q is the angle between the slats of the louvre (assuming that the slats are more or less plane, e.g. of rectangular shape, or if they are curved that they define a tangent at the exit on the downstream side) and the normal of the louvre plane (in this example the positive x-axis). The tangent vector in the xz-plane of the plane determined by the louvre slats is given by $(\cos(q), 0, \sin(q))$. A typical value of the angle q would be -45°. This would correspond to a case where the louvre panel is a shield for rain (assuming that vertically upwards is in positive z-direction).

### 3.7.21.6 Normal vector slats negative

Cartesian coordinates of normal vector of plane determined by the louvre slats for flow in negative direction. Similar comments apply here as those given above for NORMAL_VECTOR_SLATS_POSITIVE.

In many cases this vector equals the normal vector for flow in positive direction. This is the case if e.g. the louvre slats have a rectangular shape. But if the louvre slats are e.g. V-formed the plane determined by the louvre slats for flow in negative direction is different from the plane for flow in positive direction.

---

**FLACS-CFD v22.2 User's Manual**

---

**GEXCON**
The normal vector determined by the louvre slats for flow in negative direction should lie in the plane defined by the normal vector determined by the louvre slats for flow in positive direction and the normal vector of the louvre plane. If for example the louvre panel is oriented in x-direction and the normal vector of the plane determined by the louvre slats for flow in positive direction lies in the xz-plane, then the normal vector for flow in negative direction should also lie in the xz-plane.

3.7.21.7 Drag coefficient acceleration

Drag coefficient \( C_{\text{acc}} \) of drag due to acceleration of the magnitude of the velocity of the flow. This coefficient should be zero or positive. A typical value is 1 (this value is supported by theoretical considerations). Further guidance is given above.

3.7.21.8 Drag coefficient bending

Drag coefficient \( C_{\text{bend}} \) of drag due to bending of the flow. This coefficient should be zero or positive. The units of this coefficient is \( 1 / \text{radian} \). Further guidance is given above.

3.7.21.9 Drag coefficient friction

Drag coefficient \( C_{\text{fric}} \) of drag due to friction. This coefficient should be zero or positive. Further guidance is given above.

3.7.21.10 Area porosity

An area porosity of the louvre panel is specified. This is not a projected area porosity on to the louvre plane (in many cases it is not possible to see through the louvre watching normally on it, i.e. the projected area porosity is zero), but is viewed as the ratio of open area to total area in a representative cut plane parallel to the louvre plane. If e.g. the louvre consists of rectangular slats of thickness \( D \) uniformly inclined relative to the louvre plane and uniformly spaced with centre/centre distance \( 4D \), the area porosity is \( (4D - D) / 4D = 0.75 \) (this value is independent of the angle of inclination). If the frame of the louvre blocks the panel considerably, this may be taken into account. If the blockage of the frame corresponds to an area porosity \( \beta_{\text{frame}} \), the total effective area porosity in our example will be \( 0.75 \beta_{\text{frame}} \).

In the sub-grid model for louvre panels, the value of the area porosity affects the drag force only indirectly. The drag force is determined by the drag coefficients, the configuration of the louvre slats (determining the deflection of the flow downstream of the louvre), and the flow field. The value of the area porosity affects how the fluxes (of mass, momentum, etc.) are set when solving the conservation equations at the louvre plane, and this will affect the flow field.

If the profile of the louvre slats has a complicated form, it may not be obvious how to estimate the effective area porosity. If the area porosity varies in different cut-planes parallel to the louvre plane, it is expected that a cut-plane with a minimum area porosity is the most representative. Perform several simulations varying the value of the area porosity, to investigate the sensitivity of the value of the area porosity on the numerical results. The limited testing in the validation simulations reported in (Salvesen, 1996b), seems to indicate that the maximum overpressure of an explosion simulation is in general not very sensitive to changes in the value of the area porosity. In one scenario the maximum overpressure dropped about 12% when the value of the area porosity was increased from 0.6685 to 1.0.

See also:

Louvre panels can be added quickly using the Arrange items functionality.

3.7.22 Grating
Warning:

The grating model should only be used in dispersion and ventilation simulations. The current model is designed for flow through the grating, but the important effect on flame acceleration along the grating is not handled correctly. It is recommended to use a porous plate in explosion simulations.

The static pressure-loss coefficient $C_{spl}$ is defined by

$$\Delta p = 0.5 \rho |U|^2 C_{spl}$$

(3.30)

where $\Delta p$ is the pressure loss across the grating, $\rho$ is the density, and $U$ is the upstream velocity component normal to the grating. The component normal to the grating of the force per unit area from the grating on the fluid flowing through it, i.e. the drag force per unit area, is assumed to be equal to the static pressure drop over the grating. The static pressure-loss coefficient is modelled by:

$$C_{spl} = f_1(R_e)f_2(\beta)f_3(M)$$

(3.31)

where the factor depending on the geometry, characterised by the area porosity $\beta$, is given by:

$$f_2(\beta) = \frac{(1-\beta^2)}{\beta^2}$$

(3.32)

and the factor depending on the upstream Mach number $M$, and the upstream choking Mach number $M^*$ (upstream Mach no. corresponding to choking at the grating) is given by:

$$f_3(M) = \left( \frac{M^*}{M^* - M} \right)^{1/7}$$

(3.33)

The choking Mach number is related to the area porosity, $M^* = M^*(\beta)$. This relation is modelled by:

$$M^*(\beta) = 0.675\beta + 0.325\beta^4$$

(3.34)

being a curve fit to experimental data.

The factor depending on the Reynolds number, $R_e$, is modelled as a curve fit to experimental results. The characteristic length used in the Reynolds no. is the diameter of the wire (or of the rod if the grating is constructed by rods rather than wires). This curve fit and experimental data are shown in the figure below. Further details about modelling drag forces for flow through grating, are found in (Salvesen & Storvik, 1994).
3.7 Scenario menu

Figure 3.49: Factor $f_1$ related to Reynolds number

Factor $f_1(R_e)$ in pressure-loss coefficient as function of Reynolds number (logarithmic scale). Both experimental data and a curve-fit to these data are shown.

An example of a setup for a louvre panel is shown below:

| NAME       | "NoName" |
| POSITION   | 0.0, 0.0, 0.0 (m) |
| SIZE       | 0.0, 1.0, 1.0 (m) |
| MATERIAL   | RED       |
| AREA_POROSITY | 0.7   (−) |
| CHARACTERISTIC_LENGTH | 0.01 (m) |

3.7.22.1 Name

Character string identifying the grating considered.

3.7.22.2 Position

Cartesian coordinates of the corner of the grating (the corner with lowest value of the coordinate in each axis direction).

3.7.22.3 Size

The grating is assumed to be of rectangular shape. The dimension in each of the axis directions is given. One dimension should be zero, this shows how the grating is oriented. If e.g. the dimension in x-direction is zero, the normal of the grating points in x-direction. The other two dimensions should be positive.

3.7.22.4 Material

The material to be used for modelling and visualising the grating; see the section on panel materials for remarks on the usage.
3.7.22.5 Area porosity

Ratio of projected open area of the grating divided by total area. Value between 0 and 1.

3.7.22.6 Characteristic length

Characteristic length used in the Reynolds number, defined as the wire diameter (or similar dimension if it is not a wire, but a rod). A typical value of the characteristic length is 0.01 m.

See also:

Grating can be added quickly using the Arrange items functionality.

3.7.23 Gas monitor region

The GAS_MONITOR_REGION functionality is used for monitoring the amount of fuel inside a user-defined volume, e.g., a module, during a dispersion simulation. FLACS-CFD will write a text file called rt010100.FUEL which contains a number of columns showing the amount of fuel inside the defined volume. The fuel concentration range being counted towards the quantities in the rt.FUEL file can be limited by setting ER_LOW and/or ER_HIGH in the PARAMETERS namelist in a setup file.

Note:

ER_LOW and ER_HIGH do not influence the flammability range of the fuel mixture modelled in Flacs, they only limit the concentrations taken into account when reporting for a GAS_MONITOR_REGION.

The output columns Q8, Q9, [LFL:UFL], Q6/Q7 are used for risk assessments in connection to dispersion studies and estimates of explosion severity and ignition probabilities.

If more than one region is needed, then you must create a cs010100.MON file. See the section on Custom gas monitor regions for details.

3.7.23.1 FlamMass output (FLAMkg in Flacs2)

FlamMass [kg] is the mass of the fuel composition (e.g., natural gas) where the concentration of fuel (when fuel is mixed with air) is within the flammable limits (mass of air is NOT included in FlamMass).

3.7.23.2 FlamVol output (FLAM in Flacs2)

FlamVol [m³] is the volume of the fuel-air mixture where the fuel concentration is within the flammable range. This volume includes both the fuel and the air, rather than just the fuel. In other words, FLAM is the volume of the fuel-air mixture where the fuel concentration measured by the Equivalence Ratio (ER) is between ER_LOW (default: ER at LFL, ER_LFL) and ER_HIGH (default: ER at UFL, ER_UFL).

3.7.23.3 GasMass output (TOT_GASkg in Flacs2)

GasMass is [kg] the total gas mass in the mixture.

3.7.23.4 ERfacMass output (ERFACkg in Flacs2)

ERfacMass [kg] is the reactivity scaled flammable mass. It is equivalent to Q1 and evaluated as:

\[ \text{ERfacMass} = Q_1 = \sum_{i=1}^{n} \text{FUEL}_i \cdot \text{ERfac}(ER_i) \]  (3.35)

Here \( \text{FUEL}_i \) is the mass of the fuel [kg] in the control volume, and \( \text{ERfac}(ER_i) \) is defined here.
3.7 Scenario menu

3.7.23.5 Q6 and Q7 output

Q7 [m$^3$] (as function of time $t$) is defined to be the volume that has experienced a fuel concentration within the flammable range at least once during the period from the beginning of the simulation up to the time $t$. At simulation start, Q7 is set to zero; Q7 is non-decreasing as function of time, i.e. it is either constant or increasing. Q7 can be reported for

- the gas monitor region defined in the cs<job no.>.dat file; in this case, the output is stored in the file rt<job no.>.FUEL;

- a custom gas monitor region defined in a cs<job no.>.MON file; output for Q6, Q7 etc. in the monitor region named region1 will then be stored in the file rt<job no.>.MON.region01.

Q6 [m$^3$] at time $t$ is the increment of Q7 during the interval [trunc($t$) - 1 s; trunc($t$)], i.e. the last second before $t$ (trunc($t$) is the integer part of the time $t$ measured in seconds). Thus, if for example $t = 2.9$ s, this interval is [1.0 s; 2.0 s], and if $t = 3.0$ s, then the interval is [2.0 s; 3.0 s]). Q6 is often applied for ignition modelling (constant ignition sources), as it describes the increment of new volumes being exposed to flammable gas for the first time. Q6 is updated for $t = 0.0$ s, after the first time step, at $t = 1.0$ s, and then for each 1.0 s.

To explain in more detail how Q6 is determined in FLACS-CFD, let us consider a simulation starting at time 0.0 s. Then

Q6(t=0.0s) = Q7(t=0.0s) = 0.0 [m$^3$],

Q6(t equal first time step after 0.0s) = Q7(t equal first time step after 0.0s),

Q6(t=1.0s) = Q7(t=1.0s) - Q7(t equal first time step after 0.0s),

Q6(t=2.0s) = Q7(t=2.0s) - Q7(t=1.0s),

Q6(t=3.0s) = Q7(t=3.0s) - Q7(t=2.0s), etc.

If there is no time step at exactly 1.0 s, the increment of Q6 takes place at the first time step after 1.0 s (and so on). Between the increments at every full second, Q6 is reported as constant.

When a simulation is restarted from a dump file, the history for Q7 for the time period before the time step of the restart is lost. In other words, Q7 is also set to zero at the start of a restarted simulation. This means that the values of Q6/Q7 as function of time for the restarted simulation will in general not be the same as the values of Q6/Q7 as function of time for the first simulation (i.e. the simulation started from scratch, covering the first time period until the dump file is made, plus the time period of the restarted simulation reading from the dump file). For this reason, it is advised to use data for Q6/Q7 only from simulations started from scratch, not simulations restarted from a dump file.

3.7.23.6 Q8 and Q9 output

The fuel log file (rt<jobno>.FUEL) and monitor file (rt<jobno>.MON), which are output files generated for the GAS_MONITOR_REGION in the cs-file or cs-MON files, have two columns called Q8 and Q9. The quantities Q8 and Q9 are used to create equivalent stoichiometric gas clouds from real gas clouds.

Q8 This column reports the expansion-base weighted volume, that is the closed volume equivalent cloud at concentration for maximum expansion (normally near stoichiometry).

Q9 This quantity gives an improved version of the previously used Q5, where both maximum flame speed and maximum expansion are taken into account.

For enclosed situations, and situations where combustion is much quicker than venting (including quasi-detonation and flames faster than speed of sound ahead), Q8 will be a recommended equivalent cloud size.

For well-vented situations Q9 is generally recommended as equivalent stoichiometric cloud.

Q8 is defined as

$$Q8 = \frac{\sum_{i=1}^{n} V_i [Ve(ER_i) - 1]}{\max \{ [Ve(ER) - 1] : ER_{LFL} \leq ER \leq ER_{UFL} \}} \quad (3.36)$$
and $Q_9$ as
\[
Q_9 = \frac{\sum_{i=1}^{n} V_i [V_e(ER_i) - 1]ER_{fac}(ER_i)}{\max\{[V_e(ER) - 1]ER_{fac}(ER) : ER_{LFL} \leq ER \leq ER_{UFL}\}}
\]  
(3.37)

The summation $i = 1, \ldots, n$ is over all the control volumes of the numerical grid inside the gas monitor region considered where the Equivalence Ratio (ER) is between the value of ER at the Lower Flammability Limit (LFL) and the value of ER at the Upper Flammability Limit (UFL), that is $ER_{LFL} \leq ER \leq ER_{UFL}$. Here $V_i$ is the volume [m$^3$] open for fluid flow in the control volume (the volume porosity is taken into account, so only the non-blocked volume counts).

The function $ER_{fac}(ER)$ [-] has a value between 0 and 1, depending on the value of ER:
\[
ER_{fac}(ER_i) = \frac{S_L(ER_i)}{\max\{S_L(ER) : ER_{LFL} \leq ER \leq ER_{UFL}\}}
\]  
(3.38)

with $S_L$ the laminar burning velocity (corrected for flame wrinkling/Lewis number effects). For $ER = ER_{LFL}$ or $ER = ER_{UFL}$, $ER_{fac}(ER)$ has value 0; for $ER = ER_{top}$, $ER_{fac}(ER)$ has the value 1. $ER_{top}$ is in general close to, but not exactly equal to 1.0 ($ER = 1.0$ corresponds to the stoichiometric condition). In summary, in the flammable range, $ER_{fac}(ER)$ is the laminar burning velocity profile scaled to the maximum of one, while it is zero outside the flammable range.

The maximisation in the denominator, $\max\{[V_e(ER) - 1]ER_{fac}(ER) : ER_{LFL} \leq ER \leq ER_{UFL}\}$, normally occurs for a value of the Equivalence Ratio close to, but not exactly equal to 1.0 since it also includes the volume expansion ratio at constant pressure, $V_e$ [-], which depends on the Equivalence Ratio (concentration of fuel in the fuel-air mixture), and is defined by:
\[
V_e = \frac{V_{burnt}}{V_{unburnt}}
\]  
(3.39)

Here $V_{unburnt}$ is the volume [m$^3$] of the fuel-air mixture before any combustion, and $V_{burnt}$ is the volume of the combustion products; combustion is assumed to be at constant pressure, with adiabatic expansion, and at thermodynamic equilibrium (these are theoretical assumptions when defining $V_e$ that will not normally be valid during a gas explosion). By employing the ideal gas law one obtains:
\[
p V_{unburnt} = \frac{m}{M_{unburnt}}RT_{unburnt} \quad \text{and} \quad p V_{burnt} = \frac{m}{M_{burnt}}RT_{burnt},
\]  
(3.40)

where $p$ [Pa] is the pressure, $R$ is the ideal gas constant ($8314.3$ [J/kmol·K]), $T$ [K] is the temperature, and $M$ [kg/kmol] is the mean molecular weight for the mixture considered. Since the mass $m$ is the same before and after burning, the volume expansion ratio at constant pressure, $V_e$, can be rewritten as:
\[
V_e = \frac{T_{burnt}/M_{burnt}}{T_{unburnt}/M_{unburnt}}.
\]  
(3.41)

### 3.7.24 Custom gas monitor regions

Multiple gas monitor regions can be specified by using the Monitor file (cs.MON). The monitor file can either be created directly in a text file editor or in the Custom gas monitor regions section of the scenario menu. Monitor regions can be of type Line or Volume. Each gas monitor region must have a unique name `<name>`. Overall output for all monitor objects is written to a text file with the name `rt<jobnumber>.MON`, and detailed output per monitor object is written to text files called `rt<jobnumber>.MON.<name>`. A maximum of 100 monitor objects can be specified in the `cs<jobnumber>.MON` file.

#### 3.7.24.1 Line monitors

Line monitors can be used to measure fuel concentrations or pressure along a given line in the domain. The line is defined by its start and end position and will be divided into a number (default 100) of segments of equal length. Values are obtained at all intermediate points along the line by using tri-linear interpolation of
the 3D data. Detailed output for each of the points along the line can be obtained and will be written to the line monitor's specific file. In the general rt<jobnumber>.MON file, basic output for the line monitor is recorded, for example the average or integral value over the length of the line.

See also:

The syntax for the line monitor specification is explained in the monitor file section.

### 3.7.24.2 Volume monitors

Volume monitors can be used to measure amounts of fuel inside a given volume in the domain. The volume is defined by two diagonally opposed corner points of a rectangular box, start and end, resulting in a cuboid whose faces are parallel to the grid planes. The results are written to the common rt<jobnumber>.MON file and to the monitor's specific file. Several output variables are included, e.g. total amount of fuel (kg) and equivalent cloud sizes (m3). The total fuel amount is calculated by summing up contributions from each cell inside the volume.

See also:

More details on volume monitors can be found in the monitor file section. If only a single monitor region is needed then it can be specified as GAS_MONITOR_REGION in the cs-file.

### 3.7.25 User species

Gases not defined in FLACS-CFD can be defined manually as user species. A user species is defined by a number of parameters that are used in the material property models in FLACS-CFD. It is important that the values provided for these parameters are appropriate to the range of temperatures that occur in the simulated physical process.

Properties of predefined species in FLACS-CFD can be found by running a utility program called listspecie. For METHANE, the output looks like

```bash
> run listspecie METHANE
FLACS listspecie (version 1.4, 2013-12-09)
Copyright 2013, Gexcon AS
Values for METHANE
WFUEL  0.1604303E+02
LDENS  0.4250000E+03
SLR    0.1600000E+01
SLP    0.1500000E+00
SLA    0.1668000E+01
SLB    -0.3930000E+00
AMOL   0.2000000E+01
CMOL   0.1000000E+01
HMOL   0.2000000E+01
SMOL   0.0000000E+00
OXY    0.0000000E+00
AENT   0.1200000E+04
BENT   0.3400000E+01
DENT   0.5190000E+07
ASEIGMA 0.3280000E-01
BSIGMA -0.1790000E-03
AVAPPR 0.2011707E+02
BVAPPR 0.8978400E+03
DVAPPR -0.7160000E+01
VISC0  0.1141400E+03
VISC00 0.2291233E+02
VISC0A 0.2065000E-05
VISC0B 0.3039000E-07
COND0  -0.7636000E-02
COND00 0.1386000E-03
```
By just giving the command:

> run listspecie

you get the output:

FLACS listspecie (version 1.4, 2013-12-09)  
Copyright 2013, Gexcon AS

usage: listspecie species

where species can be

METHANE
ACETYLENE
ETHYLENE
ETHANE
PROPYLENE
PROPANE
BUTANE
PENTANE
HEXANE
HEPTANE
OCTANE
NONANE
DECANE
HENDECANE
DODECANE
HYDROGEN
CO
H2S
H2O
CO2
CL2

The parameters of the material property models for new species are given in the table below together with information for which applications they need to be specified.

Table 3.31: Species parameter definition depending on the type of scenario.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Unit</th>
<th>Description</th>
<th>Dispersion</th>
<th>Combustion</th>
<th>Pool evaporation</th>
</tr>
</thead>
<tbody>
<tr>
<td>WFUEL</td>
<td>kg/kmol</td>
<td>Molar mass of specie. (Same as WSPEC in python api)</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>AENT</td>
<td>J/(kg*K)</td>
<td>Enthalphy constant A for gas</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>BENT</td>
<td>J/(kg*K²)</td>
<td>Enthalphy constant B for gas</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>DENT</td>
<td>J/kg</td>
<td>Enthalphy constant D for gas</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>AMOL</td>
<td>-</td>
<td>Stoichiometric mole ratio O2/FUEL</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>CMOL</td>
<td>-</td>
<td>Number of moles C in 1 mole FUEL</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>HMOL</td>
<td>-</td>
<td>Number of moles H2 in 1 mole FUEL</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SMOL</td>
<td>-</td>
<td>Number of moles S in 1 mole FUEL</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>Unit</td>
<td>Description</td>
<td>Dispersion</td>
<td>Combustion</td>
<td>Pool evaporation</td>
</tr>
<tr>
<td>----------</td>
<td>--------------</td>
<td>--------------------------------------------------</td>
<td>------------</td>
<td>------------</td>
<td>------------------</td>
</tr>
<tr>
<td>OXY</td>
<td>-</td>
<td>Number of moles O2 in 1 mole FUEL</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>SLR</td>
<td>-</td>
<td>SL dependency on flame radius</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLP</td>
<td>-</td>
<td>SL dependency on pressure</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLA</td>
<td>-</td>
<td>SL dependency on initial temperature</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SLB</td>
<td>-</td>
<td>SL dependency on initial pressure</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LDENS</td>
<td>kg/m³</td>
<td>Density of specie in liquid state</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ALENT</td>
<td>J/(kg*K)</td>
<td>Enthalpy constant A for liquid</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BLENT</td>
<td>J/(kg*K²)</td>
<td>Enthalpy constant B for liquid</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DLENT</td>
<td>J/kg</td>
<td>Enthalpy constant D for liquid</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ASIGMA</td>
<td>N/m</td>
<td>Surface tension constant A</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BSIGMA</td>
<td>N/(m*K)</td>
<td>Surface tension constant B</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AVAPPR</td>
<td>N/m²</td>
<td>Vapour pressure constant A</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BVAPPR</td>
<td>(N/m³)*K</td>
<td>Vapour pressure constant B</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DVAPPR</td>
<td>K</td>
<td>Vapour pressure constant D</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T_MIN_VA</td>
<td></td>
<td>Lower temperature limit for vapour</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T_MAX_VA</td>
<td></td>
<td>Upper temperature limit for vapour</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VISCB</td>
<td>(N/m³)*s</td>
<td>Viscosity constant B</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VISCT0</td>
<td>K</td>
<td>Viscosity constant T0</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VISC&amp;A</td>
<td>(N/m³)*s</td>
<td>Viscosity constant A</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VISC&amp;B</td>
<td>N<em>s/(m²</em>K)</td>
<td>Viscosity constant B</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COND&amp;A</td>
<td>(J/s)/(m*K)</td>
<td>Conductivity constant A</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COND&amp;B</td>
<td>(J/s)/(m*K²)</td>
<td>Conductivity constant B</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COND&amp;A</td>
<td>(J/s)/(m*K)</td>
<td>Conductivity constant A</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COND&amp;B</td>
<td>(J/s)/(m*K²)</td>
<td>Conductivity constant B</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCRIT</td>
<td>K</td>
<td>The fluid's critical temperature point</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCRIT</td>
<td>N/m²</td>
<td>The fluid's critical pressure point</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In addition to all above properties, if the gas is flammable, and you are running an explosion scenario or a dispersion scenario where you are interested in the flammability limits or flammable volume, the gasdata file must be provided to the simulator for each flammable species. FLACS-CFD comes with gasdata files of all the species named in CASD and DIPPR species that have Gibbs data installed in the gasdata directory of the package. However, if the gasdata file is not provided for the species, it will be treated as inert and the results will be incorrect. The gasdata file contains information on the laminar burning velocities (SLAM) as a function of the equivalence ratio (ER). The first and last entries of the gasdata file must have SLAM=0, and are used as the lower and upper flammability limits.

### 3.7.25.1 Defining new species in CASD

To define a new species, open the Gas composition and volume section in the scenario menu. In the Volume fractions part you may define the volume fraction for e.g. USERSPEC_1 (double-click on Volume fractions to see the species list). Open the section User species of the scenario menu and add a new species. Then edit the new species according to the Species parameter definition table: in this case, the name should be set to USERSPEC_1 and all relevant parameters should be assigned values.

The parameters from the table are presented briefly in Species parameters relevant to all FLACS-CFD applications, together with the models that use them.

Grouping and sorting options are not allowed in this section to maintain the same sequence as in Volume fractions.

### 3.7.25.2 DIPPR

Instead of defining your own species manually, the AIChE DIPPR 801 database can be used to create FLACS-CFD species. DIPPR is an industry standard chemical component database with 2130 different compounds. To create a new FLACS-CFD species using DIPPR you must open the DIPPR dialog by selecting the Add species from DIPPR database option in the context menu of the User species section. This will allow you to pick one of the chemicals defined in the DIPPR database, for example Isopentane, and convert it into a FLACS-CFD species by pressing OK. The version of the DIPPR database used can be chosen at the top of the dialog.

How well the chemicals can be converted to a FLACS-CFD species depends on the representation of it in the DIPPR database. This can be verified by selecting the chemical and clicking on the Verify Plots button. This will open up a new dialog that shows the FLACS-CFD species properties for the selected chemical and how well the curve fitting went for the temperature dependent properties.
In order to provide SLAM data for a custom species based on a DIPPR chemical it is necessary to generate a matching gasdata file, which can be generated for the selected chemical by using the Export Gas Data File option from the DIPPR dialog. This opens the dialog shown in the figure below.

![Figure 3.52: Export gas data dialog for Isopentane](image)

Remember to rename the gasdata file to the respective species name (for example "USERSPEC_ISOPENTANE") and to set GASDATA_MODEL in setup file to the file location if it is not located in the same folder as the scenario file. When in doubt, follow instruction and examples in The JOBSPEC namelist.

In the export dialog the blue dots in the plot shows laminar burning velocities as a function of equivalence.
ratios taken from (Gibbs and Calcote, 1959), hereafter referred to as gibbs data. The gibbs data is used in conjunction with the lower and upper flammability limits. When the dialog is opened the lower and upper flammability limits are automatically calculated and set based on data from the DIPPR database.

Note:

Export Gas Data File is only available for chemicals that have GIBBS data; the chemicals that have GIBBS data can be found by looking at the SLAM DATA column in the DIPPR dialog.

3.7.25.3 Species parameters relevant to all FLACS-CFD applications

The parameters that are required even when only dispersion of the gaseous phase is considered (i.e. no combustion and no modelling of evaporation from the liquid phase) are explained below.

3.7.25.3.1 Molar mass

WFUEL [kg/kmol] Molar mass of the species. (Named WSPEC in the python api)

3.7.25.3.2 Specific enthalpy and heat capacity at constant pressure

The specific enthalpy $H_{\text{gas}}$ [J/kg] for each species is given as:

$$H_{\text{gas}} = \left(A_{\text{ENT}} + B_{\text{ENT}} \cdot T\right) \cdot T - D_{\text{ENT}}$$

(3.42)

where

$A_{\text{ENT}}$ [J/(kg K)] First enthalpy constant.

$B_{\text{ENT}}$ [J/(kg K^2)] Second enthalpy constant.

$D_{\text{ENT}}$ [J/kg] Third enthalpy constant.

The specific heat capacity at constant pressure $C_p$ [J/(kg K)] is modelled as function of temperature $T$ [K] by the linear relation:

$$C_p = A_{\text{ENT}} + B_{\text{ENT}} \cdot T$$

(3.43)

The linear approximation of the heat capacity as function of temperature is done so that $C_p$ is well approximated in the temperature range relevant for the physical process that is simulated by the core simulator flacs.

If measured data for $C_p = C_p(T)$ is found from literature in the above form, then the constant $D_{\text{ENT}}$ can be set so that the expression for the specific enthalpy $H$ [J/kg] at $T = 298.15K$, equals the standard enthalpy of formation (measured in [J/kg]). Thus $D_{\text{ENT}}$ is set to:

$$D_{\text{ENT}} = \left(A_{\text{ENT}} + B_{\text{ENT}} \cdot 298.15K/2\right) \cdot 298.15K - H_{\text{formation}}$$

(3.44)

where $H_{\text{formation}}$ [J/kg] is the standard enthalpy of formation for the species. The standard enthalpy of formation (sometimes also called standard heat of formation) is here defined as the change of enthalpy that accompanies the formation of one mole of a substance in its standard state from its constituent elements in their standard states. By standard state is here meant the most stable form of the element at pressure 100kPa and temperature 298.15K. In FLACS-CFD the units [J/kg] are used for enthalpy, other units of energy per amount of substance, e.g. [kJ/mol], may be found in the scientific literature.

To find the constants for approximating the specific enthalpy $H_{\text{gas}}$ for a species by a second order polynomial you can use a spreadsheet processing software with trendline function. However always make sure to get the dimensions correct.
3.7.25.4 Species parameters related to combustion

Besides the property models and parameters discussed in the previous section, the following ones apply when simulating combustion and explosion scenarios. Note also that modifying laminar burning velocities in gasdata file for explosion and fire simulations, will in general be needed when modelling combustion of a user-defined species (e.g. in gasdata file USERSPEC, cf. example for HEXANE here). In general the user is advised to be very careful in using user-defined species for explosion simulations, without proper validation. For screening purposes, it might be a better alternative to approximately represent your species using instead a conservative, validated species (e.g. one of the pre-defined species).

3.7.25.4.1 Fuel composition

**AMOL [-]** Stoichiometric mole ratio O2/FUEL.

**CMOL [-]** Number of moles C in 1 mole FUEL.

**HMOL [-]** Number of moles H2 in 1 mole FUEL.

**SMOL [-]** Number of moles S in 1 mole FUEL.

**OXY [-]** Number of moles O2 in 1 mole FUEL.

Here FUEL is the species considered. When for example the species is the predefined FUEL propane (C3H8), the parameters are given by:

- AMOL : 0.5000000E+01
- CMOL : 0.3000000E+01
- HMOL : 0.4000000E+01
- SMOL : 0.0000000E+00
- OXY : 0.0000000E+00

3.7.25.4.2 Laminar burning velocity Species parameters related to the laminar burning velocity SL [m/s] are:

**SLR [-]** SL dependency on flame radius

\[ SL_{after} = SL_{before} \times (P/P_0)^{SLP} \]  \hspace{1cm} (3.45)

**SLP [-]** SL dependency on pressure:

\[ SL_{after} = SL_{before} \times \left( \frac{T_0}{298.15K} \right)^{SLA} \]  \hspace{1cm} (3.46)

**SLA [-]** SL dependency on initial temperature \( T_0 \):

\[ SL_{after} = SL_{before} \times \left( \frac{P_0}{10^5Pa} \right)^{SLB} \]  \hspace{1cm} (3.47)

Here \( SL_{after} \) is the modified value of the laminar burning velocity (SL) after the dependency is taken into account (and \( SL_{before} \) is the value of the laminar burning velocity before the dependency is taken into account).

3.7.25.5 Species parameters related to pool evaporation

If also the liquid phase of a fuel is considered, for example evaporation from a liquid pool, some additional parameters are relevant, which are described here.
3.7.25.5.1 Liquid density

**LDENS [kg/m³]**  Liquid density of FUEL.

3.7.25.5.2 Specific enthalpy of the liquid phase  The specific enthalpy \( H_{\text{liquid}} \) [J/kg] is given as:

\[
H_{\text{liquid}} = (A\text{LEN} + B\text{LEN} \cdot T/2) \cdot T - D\text{LEN}
\]  \hspace{1cm} (3.48)

A common approximation is to set \( B\text{LEN} = 0 \), reducing \( H_{\text{liquid}} \) to vary linearly with temperature. The constant \( A\text{LEN} \) is given as \( C_p,\text{liquid} \) (which can be assumed to be constant for the liquid). Knowing the specific enthalpy of the gas, \( H_{\text{gas}} \), and the heat of evaporation \( dH_{\text{evap}} = H_{\text{gas}} - H_{\text{liquid}} \), the constant \( D\text{LEN} \) in the expression for \( H_{\text{liquid}} \) can be calculated. Values for \( dH_{\text{evap}} \) can be found in textbooks or online databases.

The specific heat capacity at constant pressure \( C_p,\text{liquid} \) [J/(kg K)] can also be modelled as function of temperature \( T \) [K] by the linear relation:

\[
C_p,\text{liquid} = A\text{LEN} + B\text{LEN} \cdot T
\]  \hspace{1cm} (3.49)

where

- **ALEN [J/(kg K)]**  First enthalpy constant for the species in liquid phase.
- **BLEN [J/(kg K²)]**  Second enthalpy constant for the species in liquid phase.
- **DLEN [J/kg]**  Third enthalpy constant for the species in liquid phase.

The linear approximation of \( C_p,\text{liquid} \) as function of temperature has to result in a good approximation of the heat capacity in the temperature range relevant for the physical process that is simulated by Flacs.

3.7.25.5.3 Surface tension  The surface tension \( \sigma \) [N/m] between the liquid and its vapour is modelled as function of temperature by the linear relation:

\[
\sigma = A\text{SIGMA} + B\text{SIGMA} \cdot T
\]  \hspace{1cm} (3.50)

with the constants

- **ASIGMA [N/m]**  First surface tension constant.
- **BSIGMA [N/(m K)]**  Second surface tension constant.

The linear approximation of \( \sigma \) as function of \( T \) has to be sufficiently accurate in the temperature range relevant for the process that is simulated by Flacs.

3.7.25.5.4 Vapour pressure  The vapour pressure \( p_{\text{vapor}} \) [N/m²] is modelled as function of temperature \( T \) [K] by:

\[
p_{\text{vapor}} = \exp \left( A\text{VAPPR} - \frac{B\text{VAPPR}}{T + D\text{VAPPR}} \right)
\]  \hspace{1cm} (3.51)

with

- **AVAPPR [N/m²]**  First vapour pressure constant.
- **BVAPPR [N/(m² K)]**  Second vapour pressure constant.
- **DVAPPR [K]**  Third vapour pressure constant.

If a parameter similar to \( A\text{VAPPR} \) is found for a parameterisation of the vapour pressure in [mmHg] instead of in [Pa], then one must add \( \ln(133.3224) = 4.89277025508 \) to obtain the correct value for \( A\text{VAPPR} \) when \( p_{\text{vapor}} \) is in [Pa] as is the case in FLACS-CFD. Here it is assumed that 1mmHg (mercury) is equal to 133.3224Pa (the so-called conventional millimetre of mercury).

To complete the equation for vapour pressure, the lower and upper temperature limits must be specified, \( T_{\text{MIN,VA}} \) and \( T_{\text{MAX,VA}} \), respectively. These limits will be used by the simulator to issue a warning in case the temperatures attained in the simulations leave the valid range when calculating vapour pressure.
3.7 Scenario menu

3.7.25.5 Liquid viscosity The liquid viscosity $VISC$ [Pa s] is described by the formula

$$VISC = 10^{VISC_B \left(1/T - 1/VISC_T0\right)}$$  (3.52)

with

$VISC_B$ First viscosity constant.

$VISC_T0$ Second viscosity constant.

3.7.25.6 Gas viscosity The gas viscosity $VISC_{gas}$ [Ns/m$^2$] is given by the formula:

$$VISC_{gas} = VISC_{GA} + VISC_{GB} \cdot T$$  (3.53)

where

$VISC_{GA}$ [(N/m$^2$) s] First viscosity constant.

$VISC_{GB}$ [N s/(m$^2$ K)] Second viscosity constant.

3.7.25.7 Gas thermal conductivity The thermal conductivity $COND_{gas}$ [W/(mK)] of the gas phase is modelled as

$$COND_{gas} = COND_{GA} + COND_{GB} \cdot T$$  (3.54)

with the parameters

$COND_{GA}$ [(J/s)/(m K)] First conductivity constant.

$COND_{GB}$ [(J/s)/(m K$^2$)] Second conductivity constant.

3.7.25.8 Liquid thermal conductivity Analogously to the gas phase, the conductivity of the liquid phase $COND_{liq}$ [W/(mK)] is modelled by:

$$COND_{liq} = COND_{LA} + COND_{LB} \cdot T$$  (3.55)

where $COND_{LA}$ and $COND_{LB}$ have the same meaning and units as $COND_{GA}$ and $COND_{GB}$ of the gas thermal conductivity.

3.7.25.9 Critical point temperature and pressure The temperature and pressure of the fluid's critical point are denoted $TCRIT$ [K] and $PCRIT$ [N/m$^2$], respectively.

Data for the property constants can, for example, be obtained from NIST, Reid et al. (1977), or various databases.

3.7.25.6 Example: definition of a new species, dispersion of fluorine

In this example, it is assumed that only dispersion of gaseous fluorine, $F_2$, is considered with no combustion and no evaporation from liquid.

Define a new species for $F_2$ by using the parameters for $CO_2$ except that the values for WFUEL, AENT, BENT, and DENT are changed (see the section above) to:

WFUEL 0.379968E+02
AENT 0.788386E+03
BENT 0.172791E+00
DENT 0.242737E+07

The parameters for $CO_2$ are listed by giving the command:

```>
run listspecie CO2
```
You may also change the parameters for CMOL and OXY so that they are correct for $F_2$ (but these are not essential when there is no combustion during the simulation):

\[
\begin{align*}
\text{CMOL} & = 0.000000E+00 \\
\text{OXY} & = 0.000000E+00
\end{align*}
\]

Then you can run a dispersion simulation with $F_2$ as a user-defined species. There is also an alternative approach for dispersion of $F_2$: If you find it acceptable that the enthalpy is modelled only approximately correct, then you may combine ordinary fuel components so that the mean molecular weight of the fuel composition is $0.379968E+02$, the value for $F_2$. The fuel composition will then behave approximately as $F_2$ in the dispersion simulation. It must be stressed that no modelling of combustion or evaporation can be carried out by this approach.

### 3.7.26 Radiation

#### 3.7.26.1 Radiation model

It is possible to choose the radiation model for fire or gas explosion scenarios. There are following options available:

- **Discrete Transfer Method** (default for fire)
- **Six-Flux**
- **None**

The Discrete Transfer Method (DTM) model is the most accurate model and is the default option for fire solver. This model uses an advanced DTM ray-tracing method to calculate radiation levels in the 3D domain as well as for selected monitor points. The DTM calculations can take a significant part of the total calculation time, especially when running the DTM model on the full domain, and they also increase the memory footprint of the simulation considerably. However, in many cases it is possible to limit the extent of the DTM model to a smaller part of the full CFD calculation domain, either with or without the far field model in the rest of the domain. The reduction in the DTM domain size reduces the simulation time and memory footprint significantly.

For smoke dispersion modelling, the DTM model with low number of rays (e.g. 12) will be faster and sufficiently accurate for defining flame shape and size. Six-Flux model is recommended for fire or gas explosion scenario where heat flux output is not required.

It is also possible to completely disable the radiation model by selecting None, however this is not recommended, as not accounting for any radiation will make the results inaccurate.

#### 3.7.26.2 Emissivity

**Note:**

Only available if Show advanced is checked.

The emissivity of walls and surfaces is defined with the emissivity option. This option influences for all surfaces how much of the incident heat radiation is radiated outwards again. A completely black body has an emissivity of 0 while a perfectly white body has a value of 1.0. The default value is 0.85 and is applicable for most steel surfaces. It is currently only possible to define a single value for all surfaces.

#### 3.7.26.3 ABSORPTION_COEFFICIENT_MODEL
Note:
Only available if Show advanced is checked.

This allows to select a specific Weighted Sum of Grey Gas model (WSGGM) which models how much of the heat radiation is absorbed and emitted by the gas. It is also possible to prescribe a specific constant coefficient.

- **Constant**: User-defined constant absorption coefficient.
- **Basic**: Uncoupled Total Emissivity Model of Hottel (1954).
- **Danish–WSGGM**: Uncoupled WSGGM based on (Yin et al., 2013).
- **Swedish–WSGGM**: Uncoupled WSGGM based on (Johansson et al., 2011).
- **Danish–Coupled–WSGGM** (default): Coupled/line of sight WSGGM based on (Yin et al., 2013).
- **Swedish–Coupled–WSGGM**: Coupled/line of sight WSGGM based on (Johansson et al., 2011).

It is recommended to use the Danish–Coupled–WSGGM for DTM calculations. Note the naming (e.g. Swedish and Danish) is used to indicate the university group(s) where the models were initially developed.

### 3.7.26.4 ABSORPTION_COEFFICIENT

Note:
Only available if Show advanced is checked.

Define the constant value of the absorption coefficient. Mainly used for validation purposes. 0.0 (default)

### 3.7.26.5 SCATTERING_COEFFICIENT

Note:
Only available if Show advanced is checked.

Define the value of the scattering coefficient. It is generally recommended to use the default value 0, as there is very little effect on thermal radiation. 0.0 (default)

### 3.7.26.6 MEAN_BEAM_LENGTH_MODEL

Note:
Only available if Show advanced is checked.

Define the beam length model used in the uncoupled WSGGM to pre-compute an absorption coefficient. Not used in the (default) coupled WSGGM model.

Constant value proportional to the ratio of volume to surface area.

- **Domain based** (default): Based on the whole domain.
- **Cell based**: Based on the local cell size.
- **Max ray length**: Based on the maximum length of the domain.
- **LSLIM**: Based on the characteristic geometrical dimension (LSLIM).
3.7.26.7 RADIATION_START
Note:

Only available if Show advanced is checked.

Controls the start of the radiation calculations.

- **Immediately**: Starts when the simulations starts.
- **When ignited**: Starts after ignition. (Recommended setting)
- **Absolute time**: User-defined time (in sec).
- **Relative time of ignition**: User-defined time relative to time of ignition (in sec).

3.7.26.8 RADIATION_START_TIME
Note:

Only available if Show advanced is checked.

Used in combination with RADIATION_START to define the (absolute or relative) time (in seconds) when to start radiation calculations. The default value is 0.0.

3.7.26.9 RADIATION_START_RAMP
Note:

Only available if Show advanced is checked.

To provide more stability, it is possible to enable (set to 1) a ramp function to dampen steep gradients in the source term immediately after starting the DTM calculations. The default value is 0 (i.e. disabled).

3.7.26.10 DTM_LOOP_MAX
Note:

Only available if Show advanced is checked.

Maximum number of iterations inside the DTM routine. The default value is 25.

3.7.26.11 DTM_TOLERANCE
Note:

Only available if Show advanced is checked.

Maximum normalised change in the surface radiation intensity from one DTM sweep to the next. This determines when the radiation field is considered converged. The default value is 0.001.

3.7.26.12 DTM_MOD_ITER

The frequency (in terms of time steps) with which the radiation terms are updated as the continuous phase solution proceeds. The default value is 100. For highly transient fires it is recommended to carry out sensitivities with lower values.

3.7.26.13 DTM_MOD_TIME

Time between calls to the radiation solver (in seconds). To activate this, enter a positive real value. The default value is ~1 (i.e. disabled). When DTM_MOD_TIME has a positive real value, it overrides the DTM_MOD_ITER setting.
3.7.26.14  DTM_RAY_DISTRIBUTION
Note:
Only available if Show advanced is checked.

Field of view discretisation of the DTM rays. Defines the method of distributing the rays over the radiating surface:

• **STAGGERED** (default): ray distribution staggered in the $\phi$ coordinate direction. The result of this distribution is that each ray occupies an equal volume and gives a more uniform radiation field than the uniform distribution for a given number of rays. It is recommended to use the **STAGGERED** distribution.

• **UNIFORM**: uniform spacing in the angle of rotation ($\phi$) and angle of incidence ($\theta$).

3.7.26.15  DTM_RAYS

Number of rays being traced from each surface cell for the DTM (resolution of the hemispherical discretisation).
The default value is 108. In case of strong radiation pattern artefacts (e.g. flower pattern) increasing the number of rays to 192 or higher should reduce the artefacts.

See also:
More information on the BestPractice_OptimalCalcSpeed_NrOfRays is available in the Best practice chapter.

The number of rays is automatically rounded to the nearest value according to the following formula:

• **UNIFORM** ($N^2$): 49, 64, 81, 100, ...

• **STAGGERED** ($3 \cdot N^2$): 48, 75, 108 (default), 192, ...

where $N$ is a positive integer.

3.7.26.16  DTM_VERBOSE
Note:
Only available if Show advanced is checked.

Print the CPU-time and residual for each iteration of DTM.

• 0 (default)

3.7.26.17  DTM_FULL_Flux
Note:
Only available if Show advanced is checked.

Activates the three-dimensional radiative flux output. Default is 0 (i.e. disabled). This feature is not yet implemented in FLACS-Fire.

3.7.26.18  DTM_BUFFER_SIZE
From FLACS-CFD 21.3 FLACS onwards DTM ray segments are calculated on the fly and a DTM buffer is no longer used. **DTM_BUFFER_SIZE** has therefore been deprecated.

FLACS-CFD v22.2 User’s Manual
3.7.26.19 DTM_FLUX_CUTOFF

**Note:**
Only available if Show advanced is checked.

Define the radiative flux cut-off value (in kW/m²) for combining the DTM and far-field results within the DTM domain. Below the cut-off value, results from the far-field model are used. The default value is 10 kW/m². If the far-field model is disabled, only the DTM results are shown and this setting has no effect.

3.7.26.20 FAR_FIELD_RADIATION

**Note:**
Only available if Show advanced is checked.

Activates the multi-point source far-field radiation model. Default is 1 (i.e. enabled).

3.7.26.21 FAR_FIELD_SHADOW

**Note:**
Only available if Show advanced is checked.

Activates the blocking (shadow) effect of geometry in the far-field region. Default is 1 (i.e. enabled). Note that even when this parameter is enabled, shadow effects are still ignored for the QRADFF parameter to reduce calculation times.

3.7.26.22 DTM_DOMAIN_CONSTRAINTS

**Note:**
Only available if Show advanced is checked.

This allows to select criteria to define the DTM domain:

- **Automatic (Default):** For each DTM calculation the domain is automatically defined based on DTM_DOMAIN_TEMPERATURE_CUTOFF as a criterion. The DTM domain is dynamically resized at every DTM call.

- **Automatic_min:** The DTM calculations are performed in the user-defined domain, (defined in DTM_DOMAIN_POSITION and DTM_DOMAIN_SIZE). If a portion of the flame goes outside the user-defined limits, the domain will be extended until it covers the whole flame (based on DTM_DOMAIN_TEMPERATURE_CUTOFF). The DTM domain is dynamically resized at every DTM call. This setting can be useful to force the DTM calculation domain to extend to specific surfaces and get detailed DTM heat radiation results on those surfaces. An example would be extending the domain down towards the ground for a vertical jet fire. If the domain is extended very significantly there will be significant impact on the DTM calculation times and RAM memory usage.

- **Full:** The DTM model is used in the whole domain.

3.7.26.23 DTM_DOMAIN_TEMPERATURE_CUTOFF

**Note:**
Only available if Show advanced is checked.

Temperature (in K) cut-off value to define the DTM domain. Used in combination with DTM_DOMAIN_CONSTRAINTS (Automatic, Automatic_min) to define the DTM domain. The default value is 500 K.
3.7 Scenario menu

3.7.26.24  DTM_DOMAIN_POSITION

Note:

Only available if Show advanced is checked.

Cartesian coordinates [m] of the corner of the box-shaped DTM domain (the corner with the lowest coordinate value in each axis direction). Used in combination with DTM_DOMAIN_CONSTRAINTS (Automatic_min) to define the DTM domain.

3.7.26.25  DTM_DOMAIN_SIZE

Note:

Only available if Show advanced is checked.

The size [m] of the box-shaped DTM domain in each of the axis directions. All three values must be positive. Used in combination with DTM_DOMAIN_CONSTRAINTS (Automatic_min) to define the DTM domain.

3.7.27  Combustion

3.7.27.1  COMBUSTION_MODEL

The following combustion models can be selected:

• Eddy Dissipation Concept (EDC) (default)
• Mixed is Burnt (MIB)

It is recommended to use the EDC model.

3.7.27.2  EXTINCTION

Enable extinction in the EDC combustion model.

• 0 : Extinction disabled (default)
• 1 : Extinction enabled

The main purpose of this option is to allow flame extinction. This feature can also be used to potentially improve the modelling of the flame lift-off length for jet-fires, when used together with a user defined chemical time scale. To set the chemical time scale (in seconds) use "TAU\_CHEM=" in the KEY field in the cs-file (see Special control keys). Please contact Gexcon for guidance on which chemical time scale to use and limitations. As this feature is still under development and testing it is not recommended to enable extinction for any type of scenario.

3.7.27.3  FIRESWITCH

The FIRESWITCH changes the behaviour of the combustion model in FLACS-Fire. There are three options:

With the default setting (0) only the EDC model is used; this is recommended for immediately ignited jet and for fires where the consequences of explosion overpressures are expected to be low compared to those of the fire. When automatic selection is enabled (1), the solver will automatically select the flame regime (either the EDC or pre-mixed model, similar to the model in FLACS-GasEx) in each control volume during each time step, this allows modelling of scenarios where both a fire or an explosion occurs. While this setting should give similar results as when modelling with the GasEx model for pre-mixed clouds, it has not been fully validated and should be used with caution. The third option changes the settings in the EDC model specifically for pool fire scenarios and is the recommended setting for pool fire simulations. In summary:

• 0: EDC only (default)
• 1: Pre-mixed and EDC (selected automatically). To enable this option, both a Fire and GasEx license are required.
• 2: EDC for pool fires; extinction disabled, independent of the EXTINCTION setting

Attention:
When modelling pool fires it is important to remember to set the FIRESWITCH to 2 (i.e. EDC for pool fires)

Control variables related to coupled pool-fire modelling are given in the Pool section of the scenario file.

3.7.28 Smoke/soot

3.7.28.1 Soot model

The soot model used by FLACS-CFD can be selected with this variable. The most detailed model in FLACS-CFD is the Formation-Oxidation model. This model is based on the work by Khan & Greeves (1974), Magnussen & Hjertager (1976) and Kleiveland (2005). It accounts for both formation of soot in the flame as well as (partial) oxidation of the soot. The model includes the effect of ER, temperature and pressure on the formation of soot. The subsequent (partial) oxidation of the soot is subject to the exposure time and the available oxygen.

The soot source term $\omega_s$ in the transport equitation is sum of formation term $\omega_{s,f}$ and the oxidation term $\omega_{s,ox}$:

$$\omega_s = \omega_{s,f} + \omega_{s,ox}$$  \hspace{2cm} (3.56)

The soot formation term $\omega_{s,f}$ is written as:

$$\omega_{s,f} = \begin{cases} C_f p x_{fuel} (ER)^3 \exp \left( \frac{E_f}{RT} \right) & \text{if } 1.67 \leq ER \leq 3.00, \\ 0 & \text{otherwise}. \end{cases}$$  \hspace{2cm} (3.57)

The soot oxidation term $\omega_{s,ox}$ is written as:

$$\omega_{s,ox} = -4 \rho \epsilon \frac{Y}{k} Y_{min},$$  \hspace{2cm} (3.58)

where

$$Y_{min} = \min(Y_{soot}, \frac{Y_{O_2}}{\nu_{soot}} \frac{Y_{soot} \nu_{soot} + Y_{fuel} \nu_{fuel}}{Y_{fuel} \nu_{fuel}})$$  \hspace{2cm} (3.59)

The alternative, and simpler, fixed conversion factor model uses a fixed conversion rate to calculate soot in the combustion products. As this is a fixed factor, it depends only on the fuel burning rate and is independent of equivalence ratio (ER), temperature, time etc. However to prevent non-physical soot production in lean mixtures and for mixtures around stoichiometric ratio, soot is only produced when the ER is above 1.5.

• Formation-Oxidation (default)
• Fixed conversion factor
• None

3.7.28.2 Soot yield

When the fixed conversion factor model is used, the soot yield fraction of fuel mass must also be defined.

• 0.01 (default)

The value is only used when the fixed conversion factor soot model is selected. For typical hydrocarbons the soot yield is in the order of 1%.
3.7.29 Conduction

This option can be used to enable thermally active walls.

- None (default)
- Thermally active walls

Attention:

The conduction model in FLACS-Fire is not yet validated.

3.7.30 Solid Flame Model

The solid flame model scenario is graphical user interface used to create the HEAT file. This help to assign boxes with given convection or radiation heat flux and temperature. Solid Flame (with radiation heat flux) simulation only run the full domain DTM model to get the radiation field for that given instant and does not solve for flow. This model use flame shape, size and surface emissive power from simplified model (correlations in literature or EFFECTS or FRED) and use it to build our frozen/solid flame and then compute 3D radiation field (QRAD at surfaces & monitors) including shadowing effect on objects surrounding it.

3.8 View menu

The View menu in CASD contains commands for manipulating the view.

3.8.1 Print

The Print menu allows exporting a screenshot of the CASD window into different formats:

- Postscript
- RGB
- IV
- VRML 1.0
- VRML 2.0

Note:

This command is only available when using the deprecated OpenInventor viewer.

3.8.2 Examiner viewer and Fly viewer

The default and most widely used viewer is the Examiner viewer. The Fly viewer can be used to fly through the geometry.

3.8.3 XY View, XZ View, and the YZ Views

The option XY View and XZ View display a projection of the geometry in the XY and XZ planes respectively. The options YZ East View and YZ West View display a projection of the geometry in the YZ plane along the positive and negative Y-axis respectively.
3.8.4 3D View

The 3D View option displays a default 3D view of the geometry.

3.8.5 Axis

The Axis option turns axis display on and off.

3.8.6 Maximise

Shortcut: \texttt{SHIFT+M}

The option Maximise maximises the visible window to display the entire geometry and grid.

3.8.7 Grid display

Three mutually exclusive options are available in the Grid Display menu:

- Off: The grid is not displayed. Only the geometry is shown.
- Working Direction: The grid is displayed in the working direction only.
- All Directions: The grid is displayed in the three directions.

There is one additional option, \texttt{Superimpose on geometry}, which overlays the grid on the geometry. This feature works in conjunction with the three mutually exclusive options described above.

3.8.8 Annotation

This feature is used to show/hide all the textual annotations (for example monitor point labels) in the graphical view.

3.8.9 Draw style

Different options are available in this menu:

- Off: The geometry will not be displayed.
- Default: Geometry is drawn according to database/primitive settings.
- Opaque: All geometry will be drawn opaque.
- Semi-transparent: All geometry is drawn semi-transparent.
- Wireframe: Only the edges of the objects will be drawn.
- Wireframe and default: Combination of above descriptions.
- Wireframe and opaque: Combination of above descriptions.
- Wireframe and semi-transparent: Combination of above descriptions.
- Realistic: Enable realistic lighting.
- Use textures: Enable use of textures and normal/roughness/metalness maps on geometry.
- Use reflections: Enable reflections (only available when realistic lighting is enabled).
- Use shadows: Enable shadows (only available when realistic lighting is enabled).
- Show sky: Show skydome and use sky light instead of head light if realistic lighting is enabled.
3.8.10 LOD and Properties

Note:

This command is only available when using the deprecated OpenInventor viewer.

The LOD (Level Of Details) and properties menus control the details of the geometry displayed. In order to display geometry more efficiently, the internal representation of geometry in CASD is organised spatially, and level-of-detail (LOD) and culling are used. LOD is a method used to avoid displaying all the details on objects that are far away and not visible anyway, and culling is used to avoid displaying geometry that is outside the field of view. The spatial organisation makes these methods more efficient, as large instances can be shown with varying level of detail, and parts that are not visible can be culled faster. Unfortunately it also increases the geometry loading time as a side effect. The detail level for a part of an instance is based on the screen size of its bounding box. If it is below a specified threshold, that part is shown as just the bounding box instead of displaying the individual primitives.

The threshold levels in the View → Properties dialog specify the size of the instance in pixels on the screen when the detail level should change. Minimum LOD means the instance will not be shown at all, and full LOD means it will be shown with all details. In between full and minimum it will be represented by one or more boxes.

Culling limit is used to avoid culling simple objects. Instances where the total number of primitives is below the culling limit will not be culled. This is useful because attempting to cull many small instances can be inefficient.

3.8.11 Load filter

The load filter can be used for very large geometries, to limit loading of only a part of the geometry. Instances can be filtered on material, position and size or a combination, so you can for instance only load instances of a specific material within a region of interest.

Three kinds of filters are available:

Bounding box: applies to instances that have a bounding box that intersects a user-defined box. This means it will also apply to instances that may visibly be outside the box, but where the bounding boxes overlap.

Material: applies to instances of objects of a specific material.

Size: applies to instances with a size of the largest diagonal of the instances bounding box.

The load filter dialog is available in the Geometry menu. Click the Add button to add a new filter. Right-click in the filter properties list box to bring up the pop-up menu, and select Add → Bounding box or Add → Material to add a bounding box or material filter rule. Use the pop-up menu or double click the filter to set its properties. The bounding box can also be interactively positioned and resized by dragging its handles in the geometry view.

The filter can be set to match any rule, or all rules. For instance it can apply to objects either inside a bounding box or having a material, or objects inside a bounding box that also have a specific material. In the advanced tab you can also add filter groups. This can be used to create filters that for instance apply to objects inside a bounding box, and objects inside a different bounding box with a specific material.

3.9 Options menu

In this menu, you can set certain options regarding CASD.

3.9.1 Units

Choose between the following units for the spatial dimensions: millimetres (mm), centimetres (cm), decimetres (dm), meters (m), and inches (in); the default option is meters.
3.9.2 Preferences

You can set preferences for:

- General: porosity options and compatibility switches
- Colours: background colours for examiner viewer and fly viewer
- Performance: redraw options

3.9.2.1 General

If the option ‘Write hue in macros’ is checked, then CASD will write the hue (colour) for every primitive to the macro files. This is not compatible with FLACS-CFD versions prior to v9.1r2.

If the option ‘Show warning when creating rotated primitives’ is checked, CASD will warn you if you create a primitive that is not aligned to the coordinate axes or when you rotate an existing primitive to an angle which is not a multiple of 90 degrees.

If the option ‘Enforce project and geometry numbers in job number’ is checked, only the last 2 digits in the 6 digits job number can be freely set. The first 4 digits must set according to the project and geometry numbers in the geometry database.

Threads used by FGC specifies how many threads are used by FGC (Flacs Geometry Calculator) when it is run.

Model used by FGC allows you to choose if FGC (Flacs Geometry Calculator) should use the FLACS-CFD or CAD model when calculating porosities.

3.9.2.2 Colours

In this tab the background colour for the CASD geometry windows can be set.

3.9.2.3 Performance

The control ‘Fast redraw’ currently does not work. When nVidia graphics cards are used ‘Fast redraw’ is always on, when a different type of graphics card is used ‘Fast redraw’ is off.

If the option ‘Highlighting while in motion’ is set, selected geometry will stay highlighted while it is moved. This will reduce graphics performance.

‘Show FPS indicator’ selects whether the frames-per-second (FPS) indicator in the graphics window is shown. It can be used to check the graphics speed; a higher FPS count indicates better graphics performance.

Remarks:

Two numbers are shown: the first is the maximum frame rate that would be possible with the current scene, the second number is the current frame rate.

Using the ‘Detail level’ slider, the graphics detail level can be set. A lower detail level will increase the graphics performance and vice versa.

The option ‘Max snap points’ sets the maximum number of snap points CASD will show. A high number will reduce graphics performance. The default is 25000.
3.9.3 Plugins

A plugin is a software component that adds a certain feature to CASD and can be loaded and unloaded at runtime. Plugins can be enabled or disabled in CASD's plugins window. Most plugins depend on the Python plugin, thus disabling Python will also disable all plugins based on Python.

3.9.3.1 Creating your own plugins

If you have sufficient Python expertise, you may write your own CASD plugins. This is currently not officially supported by Gexcon, and very little documentation is available. In order for CASD to find your plugin, its filename must end with plugin.py, and it must be located in either the plugins directory, or a directory specified by the CASD_PLUGIN_PATH environment variable. See the Python API section for more information on the Python interface.

3.9.4 Windows

Under this menu entry, you can access the different CASD window types, for example the 3D view/geometry window, the scenario menu, and the different tool bar sections. If you have closed one of the windows by accident, then you can re-enable it through this menu.

Note:
Some of the window types accessible in the menu belong to FLACS-Risk and are not useful when defining single simulations in CASD.

3.9.5 Layout manager

You can rearrange the individual windows and tool bars in CASD to define layouts that best suit your preferences and work flow. The ‘Layout manager’ allows you to save, activate, and manage the different layouts you may define. The layouts are preserved for future CASD sessions.

Attention:
‘Clear layout settings’ will remove all user-defined layouts!

3.10 Macro menu (deprecated)
Warning:

The macro functionality is deprecated from FLACS-CFD 20.1 and onwards. The feature have not been removed from the product, but will not be updated or be able to support new features.

The Macro menu contains commands for running and recording macros.

- **Run**: This command processes all the commands on a specified file before turning control over to the user again. If an error occurs, the processing is interrupted.
- **Record**: This is a toggle button for turning command recording on/off. When turning recording on, the command requires a macro file name. All subsequent commands are recorded on the specified file, until the recording is turned off.
- **Write Geometry**: This command writes the macro files needed to define the open geometry, including global objects and materials. CASD asks for the path to a directory where the files are placed.

### 3.10.1 Run

To create a geometry from a set of macro files, use the Run command in the Macro menu. Alternatively use the command input to read the macro file `geometry_name.mcr`:

```
* macro run geometry_name
```

If the project or geometry already exists in the database, an error message is displayed and CASD exits from the macro.

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>geometry_name.mcr</td>
<td>Creates a new geometry</td>
</tr>
<tr>
<td>geometry_name_materials.mcr</td>
<td>Creates all materials used in the geometry</td>
</tr>
<tr>
<td>material_name.mcr</td>
<td>Creates material (one file for each material)</td>
</tr>
<tr>
<td>geometry_name_objects.mcr</td>
<td>Creates all objects used in the geometry</td>
</tr>
<tr>
<td>object_name.mcr</td>
<td>Creates object (one file for each object)</td>
</tr>
<tr>
<td>geometry_name_instances.mcr</td>
<td>Creates all assemblies/instance</td>
</tr>
</tbody>
</table>

If some objects or materials on the macro files already exist in the database, an error message is displayed, and the object/material is not overwritten.

### 3.10.2 Record

The option `Record` is used to save a macro in a `000000.caj.mcr` file, where `000000` is a given job number. The macro file is written simultaneously as you use CASD, thus this function acts like a log of the performed actions.

This file can be read as executing a macro with the option Run.

### 3.10.3 Write geometry

The Write Object command in the Macro menu writes a macro file that defines the open object. CASD asks for the path to a directory where the macro file is to be placed.

This macro file must be started in the CASD main window.

The Write Geometry command in the Macro menu causes CASD to write a complete set of macro files for the open geometry. The files include macro files which create the project, geometry, all materials needed, all objects needed in addition to the assemblies/instances. The files are listed in table Macro files created by the Write Macro command.
The macro file format is not intended as a backup format. Future versions of CASD may not be backwards compatible with the menu structure and commands in the current version.

The Copy command in the Database menu can be used to make a copy of a geometry within the same project. The macro files created by the Write Geometry command in the Macro menu can be used for copying the geometry from one database to another. They can also be used for copying one geometry into another (existing) geometry.

To copy one geometry (geo1) into another (geo2) in the same database, open geo1 and execute the Write Geometry command in the Macro menu. Exit from geo1 and open geo2. Create a new assembly and execute the macro:

```
* macro run geometry_name_instances
```

Problems occur if geo2 contains local object(s) with the same name(s) as in geo1.

### 3.11 Help menu

The purpose of the Help menu is to provide you with relevant information concerning the general use of FLACS-CFD and the active FLACS-CFD license.

#### 3.11.1 Online Help

This menu opens this manual in the online help browser.

#### 3.11.2 User's Manual (pdf)

This menu item will open up the PDF version of the FLACS-CFD User's Manual in the system default PDF viewer.

#### 3.11.3 Keyboard shortcuts

This menu opens a window summarising various controls
Figure 3.55: The CASD keyboard shortcuts window lists important keyboard and mouse shortcuts.

3.11.4 License terms
This menu opens a window that contains the FLACS-CFD license terms.

3.11.5 About CASD
This menu displays the FLACS-CFD splash screen with information about the version of FLACS-CFD and the version of CASD.

3.12 Python API
FLACS-CFD comes with a Python API. An API reference is available from the Python menu in CASD. The API is deemed stable, but is subject to change in future releases. Python scripts can be run from inside CASD using the Python plugin, or from the command line using the run_python script provided in the bin directory. For instance, to run the geometry example execute

```
> run_python python/flacs/examples/geometry_example.py
```

Several more example Python scripts are available in the bin/python/flacs/examples directory.

3.13 Tools

3.13.1 Snap points
Snap points are points on the geometry that can be activated by clicking on them. The current snap point is indicated with a gray sphere. When moving the mouse pointer over a snap point the cursor changes to a cross and you can click to activate the snap point. Snap points are available on edges, faces and vertices. You can also hold CTRL to position the snap point at any location on the geometry. Hold CTRL and click outside the geometry to disable the snap point.
Snap points are available in both geometry view and the object editor. In the geometry view snap points are only used for placing instances, and are only enabled for the currently selected instance. The default position for a new instance is the current snap point.

In object view snap points are used for placing the next primitive, as well as selecting the rotation or scale origin, or the from/to coordinate when moving primitives.

Snap points can also be used to interactively move, rotate or resize a primitive. When the properties dialog is open, click and drag snap points to edit the primitive. Snap points on endpoints of cylinders can be dragged to adjust the cylinder length and orientation, and the snap points on the midpoint of the cylinder can be used to move it. On boxes, the snap points on the edges and vertices are used to resize it, and snap points on the faces of the box are used to move it.

3.13.2 Terrain import

3.13.2.1 Digital elevation data

3.13.2.1.1 Introduction

For scenarios where the terrain may significantly influence the flow, it is important to account for the topography in the FLACS-CFD modelling. FLACS-CFD can import digital elevation maps (DEM) which can be used for modelling the effects of terrain on the simulations, and for visualising elevations both within and outside the simulation domain.

**Warning:**

Terrain import functionality was introduced in FLACS-CFD v10.5 and has some limitations.

3.13.2.1.2 DEM file formats

There is a wide variety of different DEM file formats. FLACS-CFD uses the GDAL (Geospatial Data Abstraction) library to import DEM files. This library supports 142 scalar and 84 vector DEM file formats. Since it is impossible for Gexcon to test and support all these formats, testing and support are limited to the most common format, GeoTiff. For most other formats there is no technical reason that they cannot be imported in FLACS-CFD, but Gexcon cannot give formal support if any issues are encountered.

Only a limited number of DEM formats (e.g., GeoTiff) are georeferenced, i.e., contain meta information about the horizontal origin, dimensions, scale, or units, and even GeoTiff files do not necessarily contain this information. If insufficient georeferencing data are available, the import algorithm will try to guess the unit. It is important that you verify the suggested unit and, if necessary, correct it to ensure that the terrain is imported at the correct scale.

3.13.2.1.3 Data sets

Digital terrain data are available from various sources. Two of the most comprehensive DEM sources in the public domain are the Shuttle Radar Topography Mission (SRTM) and the Advanced Space Thermal Emission and Reflection Radiometer (ASTER) projects. Initially only low resolution data were available, but gradually also higher resolution (30-90 m) data has been released. An overview of the data properties can be found in the table below.

<table>
<thead>
<tr>
<th>DEM data source</th>
<th>SRTM 3-arc-second</th>
<th>SRTM 1-arc-second</th>
<th>ASTER GDEM2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Latitude coverage</td>
<td>$60^\circ$N – $56^\circ$S</td>
<td>$60^\circ$N – $56^\circ$S</td>
<td>$83^\circ$N – $83^\circ$S</td>
</tr>
<tr>
<td>Horizontal resolution</td>
<td>3 arc-second (±90 m at the equator)</td>
<td>1 arc-second (±30 m at the equator)</td>
<td>1 arc-second (±30 m at the equator)</td>
</tr>
<tr>
<td>Vertical accuracy (95% CI)</td>
<td>± 6 m</td>
<td>no data</td>
<td>± 17 m</td>
</tr>
</tbody>
</table>
Opinions about the relative merits of the different data sources (SRTM vs. ASTER GDEM) differ. However, the consensus seems to be that the SRTM data have slightly better vertical accuracy and fewer artefacts, whereas the ASTER data cover a wider latitude range and, in some cases, are more accurate for very rugged mountainous terrain (e.g., sharp valleys and ridges). Regardless of the data source, it is important to verify that the terrain has been imported correctly.

**Attention:**
Please observe the citation requirements for all DEM data!

The US Geological survey (USGS) hosts the EarthExplorer website, which is straightforward to use and supports interactive searching and downloading of a wide range of publically available DEM data, including SRTM and ASTER GDEM (see the screenshot below). In most cases, a relatively large DEM file tile will be downloaded (e.g., 100 km x 100 km), but the cropping functionality in FLACS-CFD can be used to select a smaller section to be imported.

![Figure 3.56: The USGS EarthExplorer website.](image)

### 3.13.2.2 Terrain import in CASD

A terrain surface can be imported into CASD in the object editor by choosing File → Import Terrain. When importing a terrain to a scenario in CASD, the terrain information is resampled onto a new 2D grid and added to the list of primitives. It is possible to add multiple terrains to the scenario geometry. Saving the scenario stores the terrain primitives in the geometry file (e.g., co000000.geo), and they will be processed by FGC (Flacs Geometry Calculator) when performing the porosity calculations. When choosing Import Terrain, a dialogue is presented. After clicking Open terrain file you have to select a file containing terrain information. CASD supports a number of file formats for terrain, e.g., image files (height maps), DEM files (Digital Elevation Model), GeoTIFF and the co.tri file format used internally. If CASD supports the type of the selected file and can read the terrain, then the Import terrain dialogue displays the selected terrain file as an image in which each pixel is coloured according to height: values from low to high are represented qualitatively by the colour sequence blue - green - yellow - red.
The text line below the terrain overview displays information related to the position of the mouse in the image, showing the X and Y coordinates and the height value for that position:

- If the selected terrain file includes a valid georeference, the X and Y values represent the geographical position in the file's reference system.
- If the terrain file is a basic image file (e.g. greyscale height map) each pixel will have the default size of 1 square meter, and the X and Y position represent the pixel position within the file.

As terrain files may cover large areas with high resolution, the amount of terrain data often needs to be reduced. The Import terrain dialogue allows you to reduce the size of the terrain data to be imported in two ways:

- You can restrict the range of the terrain to be imported to a subregion of the terrain file.
- You can reduce the resolution of the imported terrain by setting a skip value.

These options are explained in the next two paragraphs.

### 3.13.2.2.1 Defining a subregion

You can define a subregion by moving and scaling the red rectangle in the terrain overview using the mouse. You move the edges of the red rectangle by pointing the mouse at an edge and then holding down the left mouse button while moving the mouse. You can also point the mouse inside the red rectangle, hold down the left mouse button and move the mouse to move the whole subregion. Based on the terrain input file and the selected subregion, CASD displays the Original and Selection size (in X and Y direction) and the resulting height range, respectively, below the terrain overview.
Attention:

Georeferenced files can represent the terrain using various units and coordinate systems. If the unit type is not detected correctly, then the resulting terrain size may be incorrect. The terrain unit is set to metres by default, but DEM and GeoTIFF files may define the terrain in arc degrees or arc seconds, and this may lead to an incorrect representation (width and height) of the terrain. You can normally detect this by looking at the original values (X- and Y-size) displayed below the terrain overview. If these values do not match the expected ranges, or if the values are very large or small, then set the Planar units to the correct unit to fix this.

3.13.2.2 Reducing the terrain resolution

The Size of grid cell in meters input box defines the resulting spacing between grid lines, i.e., the size of each grid cell. The terrain grid cell size is the same in the X and Y direction and does not need to be identical to the FLACS-CFD grid. The terrain will be resampled to the simulation grid in the porosity calculations, so there is no need for the terrain to have higher resolution than the simulation grid; in most cases it will be coarser.

When pressing OK in the Import terrain dialog, the terrain will be converted to a 2D grid (similar to a FLACS-CFD simulation grid) with the interpolated height saved for each grid point. If the number of grid cells in the terrain representation becomes too large, then the frame rate in the geometry view in CASD may be affected. A warning will be displayed upon pressing the OK button if the resulting terrain representation contains more than 1 million grid cells. A large terrain representation will also affect the porosities calculation time in FGC (Flacs Geometry Calculator), and visualisation performance in Flowvis.

Some terrain files contain height values that are below the elevation of interest, for example elevations below sea level. To discard height values below the required level, the input box labeled Discard points with height below or equal to allows you to set the minimum value. Points with height values below this minimum will be set to the given value.

Georeferenced terrain files normally have coordinates far away from the origin, which may be impractical when loading the terrain in CASD. Therefore, the Import terrain dialog contains a Translate origin of imported terrain to (0,0) checkbox, which is checked by default. When checked, the lower left corner of the terrain will be placed at the origin (0,0) for the CASD coordinate system. Clicking the OK button in the Import terrain dialogue closes the dialogue and displays the terrain as a geometry surface in the CASD geometry view.

If the coordinate system in CASD has already been georeferenced, either manually or by previously importing a georeferenced terrain, the option to “Translate terrain according to georeference” will be enabled, and makes it possible to automatically align terrain patches next to each other during import. If the coordinate system in CASD has been georeferenced the label in the lower right of the object editor window will say “Georeferenced” instead of ”Not georeferenced”.

FLACS-CFD v22.2 User’s Manual
3.13.2.3 Terrain transformations

To make the terrain match other geometry, it is possible to apply transformations like rotation, translation and scaling the same way as for the other primitives. It is also possible to set a Materials on the terrain, but mapping a real terrain texture is a bit difficult in practice as the texture has to match the terrain in both size and position.

Transformations can be used to adapt the orientation, position and scale of the terrain to match the geometry in the scenario. It can also be used to set the correct orientation and size of terrain coming from files that do not contain georeference information. For example, grayscale height maps may contain correct height values for each pixel, but not include information about the extent (i.e., the size in the X and Y directions) of each pixel. In such cases, the terrain should be scaled manually to get the correct proportions.

3.13.3 Georeferencing

Geometry and terrain can be georeferenced in CASD by choosing Geometry → Georeference. Georeferencing can be used in CASD to match the placement of geometry and terrain, and in Flowvis to export georeferenced images to the GeoTIFF file format. It is performed in CASD by defining a snap-point by CTRL+left-click on the geometry or on the terrain in the 3D View, and then open the Georeference dialog from the Geometry menu.

Figure 3.58: A terrain in the CASD geometry window.
The georeference dialog is used to map a position in the FLACS-CFD coordinate system to a position in the real world. If a snap-point was set before opening the dialog, the coordinates of the snap point will be set in the FLACS-CFD Coordinate fields. The position in FLACS-CFD coordinate is matched with a position in the real world, either by writing a LatLong/UTM coordinate in the available fields, or by clicking in the interactive map view. If an imported geometry has been rotated in order to align pipes or buildings to the coordinate axis, the same rotation can also be applied in this dialog, to match the orientation in the real world.
Clicking the OK button after setting the coordinates will compute and store the real world coordinates of the FLACS-CFD coordinate system origin, if this has not been set before, and close the dialog.

Note:

When importing a terrain, the georeference in CASD might be set automatically if the source file contains georeference information, e.g., when importing from a GeoTIFF or DEM file.

If the process of setting the real-world coordinates of the FLACS-CFD origin has already been performed, either manually or through terrain import, clicking OK will open a new dialog with four choices:

1. Move terrain
2. Move geometry
3. Move origin
4. Reset location

The three first choices will modify the coordinates of the terrain, the geometry, or both, respectively. This can be used to match the position of the item in relation to each other, and in relation to position in the real world. Choice number four, Reset location, will recompute the position of the FLACS-CFD coordinate system origin, the same way as it is done the first time the georeference process is performed.

![Set georeference location dialog](image)

Figure 3.60: The Set georeference location dialog is used to move the terrain and/or the geometry in relation to the positioned FLACS-CFD coordinate system, or to reposition the FLACS-CFD coordinate system.

An example of how to use the georeference functionality is where a height map of a terrain and a geometry model of a process plant are available. The following procedure can be used to match the position of these two items in CASD: The first step is to set the real world position of the FLACS-CFD coordinate system. This is achieved by placing a snap point on a identifiable feature in the terrain, opening the georeference dialog, finding and clicking the matching feature in the map, and click OK. Step two is to find a identifiable feature in the geometry (e.g., the corner of a building) and place a snap point on this feature, open the georeference dialog and click on the matching feature in the map view, click OK, and then select Move geometry in the Set georeference location dialog.

The geometry will now be positioned correctly in the terrain and both items will have their correct position in the real world, which can be utilized later when exporting images from Flowvis.

### 3.13.4 Plan drawings

Plan drawings can be used to place an image behind the geometry, to assist with modelling or placing instances. Click the plan drawings icon in the tool bar to open the plan drawings dialog.
Figure 3.61: Geometry with plan drawing underlay

Click the New button and select an image file of format png or jpg to create a new plan drawing. The plan drawing can be oriented along any of the three coordinate axes.

The size of a new drawing is default set to 100 x 100 meters with one corner in origo and aligned in the XY plane, but the position and size can easily be calibrated. Click the Calibrate button and position the two points at two known coordinates by dragging them. Enter the coordinates for each point and click OK. If you change the orientation, the plan drawing must be re-calibrated for it to be correct.

The elevation (z value) is set in the plan drawings main window.

If the Lock aspect ratio check box is checked then the drawing will be scaled equally in both width and
height. If it is not checked the drawing can be scaled differently in each direction if necessary, i.e. if the scale in the original drawing is different in width and height.

![Figure 3.63: The plan drawing calibration window](image)

Plan drawing settings are now saved together with the database, but in previous versions of CASD the settings were saved separately. Click the **Import** button to import previously saved settings.

Units in the plan drawings window and plan drawings calibration window is according to the chosen **units**.

### 3.13.5 Clip planes

The clip plane functionality lets you hide parts of the geometry to be able to see into obstructed areas. Six clip planes can be shown simultaneously, typically one for each of the positive and negative axis directions.

The clip plane functionality is enable and disabled by selecting the **Toggle clip planes** icon in the icon bar. To edit the clip planes click the **Edit clip planes** icon.
Groups of clip planes can be named and you can select which group of clip plane is active by using the upper menu. The clip plane groups and settings are stored in the geometry database such that they are available for later sessions and other users.

By using the menu 'Edit clip plane' the various clip planes in the group can be edited. The clip plane can be enabled and disabled, and the cut surfaces can be closed or be left open.

The menu 'Direction' sets which axis direction the cut plan will be aligned to, and whether the positive or negative side of the clip plane will be shown.

If a snap point is set, then the clip plane can be placed at this location by clicking Place at snap point.

### 3.13.6 Arrange items

The Arrange items functionality allows to quickly define a large number of items like monitor points and pressure relief panels in a defined pattern.
3.14 Working with very large geometries

Note:

The Arrange items functionality is implemented as a Python plugin which must be enabled to make the functionality visible.

![Arrange items interface](image)

Figure 3.65: Arrange items

Items can be created in the following patterns:

- three-dimensional array
- plane
- line
- custom

If no items are created when Arrange items is selected, items with default settings will be created. If an item is created beforehand and is selected whilst Arrange items is selected, then the settings for this item will be used for all created items.

Arrange items is available by MOUSE+RIGHT in the scenario sections where it is available.

3.14 Working with very large geometries

The complexity and size of geometry models increase continuously. Geometry data sets become bigger and bigger both due to an increasing level of detail, but also because FLACS-CFD is being used for facilities which are physically larger, such as land based petrochemical plants, e.g. refineries. Some users may face problems handling such very large geometries in CASD. Gexcon has significant development focus on facilitating the use of very large geometries, and both the release of the 64 bit version of CASD and the new version CASD 7 with FLACS-CFD v10.0 have been major milestones in this direction.

Still, if a geometry is very large, some tricks and techniques can help to handle the geometry smoothly and successfully; the current section contains some advice to this end. For an introduction to importing geometries from CAD software into FLACS-CFD see the section on geo2flacs.
3.14.1 Increase the stack size on Linux

On Linux, the shell assigns programs a certain stack size for saving local variables. Some grid-related data in Flacs is saved in stack variables and therefore large geometries can render the default stack size assigned by the shell insufficient. Stack overflow may be signalled by an error message like

```
Error: Simulation aborted
```

but other messages may be given, depending on the system and/or shell used. You can check the current default stack size by running

```
> limit stacksize
```

in csh or

```
ulimit -s
```

in sh/bash. To avoid stack overflow, the stack size limit may be increased or set to unlimited, which means the complete system resources may be used. To set the stack size to unlimited do

```
> limit stacksize unlimited
```

in csh (see `limit` in `man csh`) or

```
> ulimit -s unlimited
```

in sh/bash (see `ulimit` in `man bash`).

3.14.2 Filter small objects in geo2flacs

Even when a geometry seems relatively small, the detail level can be extremely large (down to nuts and bolts). Tiny objects require the same resources as big objects, which may lead to memory problems as described above.

To filter away small objects in a geo2flacs conversion of the geometry, use the geo2flacs options `-d` and `-l` which disregard objects that are smaller than the specified diameter (`-d`) or length (`-l`). Furthermore, the option `-n` can be used to group small objects into larger objects. A typical value passed to geo2flacs is, e.g., `-n10000`.

3.14.3 Load filter

Another technique that can aid when working with very large geometries is to apply a load filter. Once you have read the complete geometry into CASD, save it to a database and exit CASD. Then restart CASD and open the geometry again after setting an appropriate load filter based on position, material, size, or a combination.

3.14.4 Open Inventor viewer enhancements

If the default OpenSceneGraph (OSG)-based viewer does not work well with your data or hardware, you can try to run CASD with the older, Open Inventor-based viewer. In this case, you should enable an extension of the Open Inventor viewer by setting the environment variable `CASD_USE_NEW_INSTANCE_DRAWING`; then start CASD with the command line option `"-viewer iv"`, i.e.

- on Linux: open a shell and set the environment variable; e.g. if you use the csh/tcsh shell, type

  ```
  > setenv CASD_USE_NEW_INSTANCE_DRAWING 1
  ```

  or, if you use the bash shell, type
3.15 Known issues in CASD

> export CASD_USE_NEW_INSTANCE_DRAWING=1

Then start CASD by

> run casd

• on Windows: open a command prompt and type

> set CASD_USE_NEW_INSTANCE_DRAWING=1
> casd

Alternatively, you can set the variable via the `Environment Variables` dialogue.

3.14.5 Clip planes

You can use clip planes to hide parts of the geometry and improve performance for the remaining parts. Clip planes are supported in both the main window and the object window.

3.15 Known issues in CASD

• When using VirtualGL on linux, the scenario items menu for simulation types does not work. This can be fixed by setting the environment variable `QT_XCB_NATIVE_PAINTING=1`
Chapter 4

FLACS-CFD simulators

This chapter describes various aspects of the CFD simulators in FLACS-CFD:

• preparations before running Flacs simulations,
• how to manage and monitor simulations with the RunManager,
• how to manage and monitor simulations from the command line,
• different variants of the Flacs simulator,
• the different kinds of input files for the Flacs simulator,
• the output file types of Flacs,
• which output variables the Flacs simulator can produce.

See Getting started for a detailed description of how to install FLACS-CFD and the basic steps to get started using it. To learn more about starting simulations from the command line please refer to the sections Starting simulations and Create and use run scripts.

Preconditions:
On Linux it is recommended to define an alias for running FLACS-CFD 22.2 programs:

> alias run /usr/local/Gexcon/FLACS-CFD_22.2/bin/run

On Windows, a desktop icon for the RunManager is created during the installation of the FLACS-CFD package.

4.1 FLACS-Cloud

As a FLACS-CFD license holder, you can obtain access to the FLACS-Cloud service, which allows you to run FLACS-CFD simulations on a high performance computing (HPC) service. FLACS-Cloud gives you access to scalable computing power – all from the FLACS-CFD graphical user interfaces that you are used to: pre- and postprocessing happen just as in your normal workflow, just the computing-intensive parts are carried out on the HPC system. Scenario files are automatically uploaded and result files can be downloaded to your local system. Data transfers are through secure network connections. Key benefits of the FLACS-Cloud service:

• Highly scalable computational capacity.
• High level of security.
• No investment in in-house computing equipment required.
• Reduced lead time on projects.
• Easy to use with existing workflows.
• Attractive CPU-hour price.

Note:
To gain access to the FLACS-Cloud service get in touch with FLACS-CFD support at flacs@gexcon.com.

Security
More information is available by request to flacs@gexcon.com.

4.2 Check list for running simulations

The sequence of tasks involved in a general FLACS-CFD simulation includes:

1. stating your problem,
2. defining sensitivities, or parameter variation,
3. defining and verify the geometry,
4. defining and verify the grid,
5. calculating and verify the porosities,
6. defining and verify the scenario,
7. running the simulations,
8. checking the simulation log files for errors,
9. presenting the results,
10. storing all data for later use.

Remarks:
It is important to check the correctness of all input parameters.

Below is a recommended check-list for basic QA of the simulation set-up:

1. Avoid large Courant numbers (CFLV and CFLC).
2. Locate ignition in an unblocked control volume.
3. Locate monitors in unblocked control volumes.
4. Define realistic discharge parameters for leaks.
5. Verify vent areas.
7. Avoid strong transient wind build-up.
8. Check disk space and access rights.
9. Minimum required files:
   • grid file (cg),
   • obstruction file (co),
   • porosity file (cp),
   • scenario file (cs).
4.3 The RunManager

RunManager is a simple graphical user interface to start and monitor simulations, both on the local machine as well as in FLACS-Cloud. Running simulations from the RunManager is very efficient and provides you with the possibility to run a single simulation, or a batch of simulations, either single- or multi-threaded. To start RunManager use the following commands or click on the desktop icon.

Linux:

> run runmanager

Windows:

> "C:Program Files\Gexcon\FLACS-CFD_22.2\bin\runmanager"

4.3.1 Porosity calculations

RunManager automatically chooses whether the porosity file should be calculated using FGC. The criteria are the same as described in Simulator versions.

If the porosity file exists, but meets any of the criteria below, then it must be recalculated to start the simulation:

- Triangle file has changed since porosity was last calculated.
- Geometry file has changed since porosity was last calculated.
- Grid file has changed since porosity was last calculated.
- The scenario file has changed since porosity was last calculated such that it could have an impact on the porosity calculation.
- The model used to calculate the porosities has changed (i.e., from CAD to FLACS-CFD model). FGC only.

4.3.2 Simulation versions

RunManager uses runflacs when running simulations. The simulation variant is determined by runflacs. See Simulator versions for more information.

4.3.3 Running jobs locally

To add a simulation file to the RunManager, click on the Add Directory button on the right-side bar and browse to the relevant folder. You can add multiple directories to the RunManager window, but only one directory at a time. Adding a folder that does not contain any FLACS-CFD simulations will close the directory browser without any warnings, and without adding anything to the list of simulations.
Note:

When browsing the content of the directories is not shown, except sub directories.

Figure 4.1: Add jobs to RunManager by adding directories that contains scenarios.

You can always remove the jobs, or rescan the directory to show newly added jobs, by right-clicking on the directory and choosing the relevant function from the context menu.

Figure 4.2: Right-click on a job and choose between Remove Jobs and Rescan Directory.

To run selected simulations in the RunManager, click on check-box in the Job number column next to the the simulations you want to run and then click on the Simulate button on the right-side bar. RunManager will limit the number of scenarios to start at the same time to the number of logical cores available, but will not take into consideration if individual simulations are set up to run on multiple cores.

4.3.4 Running multiple simulations in ”Batch Run”

Running simulations in Batch Run gives more control over how many simulations to run in parallel, and which simulations to run first. Mark the simulations you want to run by ticking off the box to the left of the job number, and then click on Batch Run to open the Batch Run dialog.
4.3 The RunManager

Figure 4.3: Select simulations to run in batch mode by ticking off several jobs and click on **Batch Run**.

In the **Run FLACS-CFD jobs in batch** dialog, you will see the selected jobs in a prioritized list queued for processing. You can prioritize which jobs to run first by changing the order of jobs in the queue. Mark the job you want to move and click on the **Move up** or **Move Down** buttons to change the order.

As it is possible to start and run multiple jobs simultaneously, you can set the number of jobs to run in parallel depending on the number of CPU cores and RAM in your PC. In the example below, RunManager will start the first two simulations upon clicking on the **Simulate** button. Additional jobs in the queue will be started as soon as one of the running simulations has finished.
Figure 4.4: Edit the number of jobs to run simultaneously, and click **Simulate** to start the batch process.

After you click on **Simulate**, you can close the batch window and monitor the simulations from the main window as described in the section below.

### 4.3.5 Monitoring the simulations

In the main window of RunManager, you have a nice overview of the simulation status.

When you select a job (by clicking on it) the lower window will be updated with the corresponding log file or plot showing parameters of interest. Choose the **Log file** tab or the **Plot** tab to switch between the different views.
4.3 The RunManager

4.3.6 Setting job parameters

Selecting a job and clicking Parameters, opens a dialog where you can adjust the simulator settings. Checking the cc File or the Setup file check boxes enables the Edit buttons that open a text editor window for these files. The Renice check box makes it possible to set the priority of the simulation process if you are running on the Linux operating system. On Windows the renice setting has no effect. The Simulator option check box enables a text field where you can add simulator options in the form “opt1=val1,opt2=val2” as you would do when starting flascfd from the command line. Enabling the Parallel run check box enables use of the parallel solver. The numerical input box next to it lets you configure how many CPU cores to assign to the simulator process.

4.3.7 Suspend, resume and abort simulations

Simulations that are running can be suspended or aborted using the action buttons in RunManager. Suspend button gracefully interrupts the process flushing all output to the output file along with all information needed for continuing the simulation later (resume). The Resume button will only be activated when selecting simulations that has previously been suspended. Abort button kills the simulation process right away and should be used only in case the user does not intend to use simulation results, as they are not guaranteed to be available in the output file.

Warning:

SUSpending and resuming simulations is only supported by flascfd; this means that simulation types that are run using flacs2.8.0+ and below will not support this feature.

Suspend/resume functionality has some known deficiencies that will be resolved in future releases:

- The temporal derivatives are not retained accurately, and therefore small differences between continuous and resumed solution should be expected.
- Some aggregated variables like TDose, QDose, Pdeath are not saved to the dump snapshot.
Running simulations can also be suspended from the command line. Simulations detached from the console or running in the background can be suspended by sending SIGINT signal on Linux or CTRL+C signal on Windows. In order to send SIGINT signal on Linux use the following command, using process id PID:

```
> kill -2 <PID>
```

Hitting Ctrl+C in the console the simulation is running will send suspend signal to the simulation on both Linux and Windows. The simulation can be resumed from the command line using option nload=0:

```
> run runflacs 010101 nload=0
```

When the nload=0 command line option is used, the simulator ignores the LOAD, JLOAD and NLOAD settings in scenario file and cc-file, and loads the latest dump snapshot from the cgns file. This allows simulations to be resumed without modifying the scenario file. Alternatively, simulations can be resumed running standard command and setting LOAD=0 and setting JLOAD to an empty string.

**Note:**
Suspend and resume does not work in conjunction with Batch run or when connected to FLACS-Cloud.

### 4.3.8 Disk Space usage

The disk space used by computed scenarios is displayed in the last column in RunManager. For scenarios that has not been simulated, the required disk space is estimated based on scenario parameters like number of grid cells, simulation duration, number of single field 3D outputs, number of monitor points and panels, MODD, NPLT, DTLPLOT, etc. For simulations that have finished, the actual used amount of disk space is displayed.

### 4.3.9 Automatic halfing of CFL numbers

If a simulation fails due to mass residual error (Mass residual) the RunManager is capable of automatically halving the CFL numbers before restarting the simulation.

If a simulation has failed due to a mass residual error, and the user tries to rerun the simulation, a dialog will be opened asking if the CFL numbers should be halved before restarting the simulation.

If the halved CLF numbers are below 5 and 0.5 (CFLC and CFLV) the values will be reset to the original value, and the simulation setup should be manually inspected in accordance with the guidance found in Mass residual.

### 4.3.10 Enabling logging in the RunManager

Logging can help finding issues related to RunManager by dumping info-, warning-, and error-messages to file.

To enable logging in the RunManager you must set the `RUNMANAGER_LOG_ENABLE` environment variable to 1. This will by default create a log-file: `runmanager.log` for information specific to the RunManager. This file can be specified manually by setting the `RUNMANAGER_LOG_FILE=<path>` environment variables.

### 4.3.11 Known issues with RunManager

- When connected to FLACS-Cloud the simulation status sometimes show the status derived from the local scenario files instead of showing the remote status if the remote status is "NOT READY", e.g., when there are no files uploaded to FLACS-Cloud.

- Mixing manual operations (e.g., starting a simulation from the command line) with operations triggered from the user interface in RunManager might lead to wrong simulation status reported by RunManager.
4.4 Managing simulations from the command line

4.4.1 Starting simulations

See the sections A Tour of FLACS-CFD and Introductory example for basic examples on how to get started using the FLACS-CFD package. The simulator can be started from the command line with the command:

Linux:

```bash
> run runflacs 010101
```

Windows:

```bash
> runflacs 010101
```

Additionally, the following command line options can be appended:

- `NumThreads=<NT>` - only if parallel version of the solver is used, `<NT>` indicates number of threads, see FLACS-CFD parallel solver
- `setup=<filename>` - run with specified setup file `<filename>`
- `nload=<NLOAD>` - enforces loading specified dump snapshot, see Suspend, resume and abort simulations
- `legacy_output=true` - enables legacy output files, including Scalar-time output file (Flacs2 only) and Field output file (Flacs2 only). Note that this option won’t be maintained and will soon be deprecated (provided only to help customers transition from Flacs2 to flacscfd)

To include multiple command line options, separate them by comma, and contain within double quotes. For example, to run parallel version on `<NT>` cores with a setup file `<filename>`, the following syntax can be used:

```bash
> run runflacs version _omp <jobnr> "setup=<filename>,NumThreads=<NT>"
```

4.4.2 Simulator versions

Runflacs will automatically choose the correct simulator variant to use, depending on the contents of the scenario file. By default, flacscfd simulator will be used.

4.4.3 Monitoring simulations

On Linux, you can use the `tail` command to monitor the progress of a FLACS-CFD simulation:

```bash
> tail -f tt010101
```

This command will continue to print out the last lines of the log file tt010101 as they are being added to the file.
4.4.4 Create and use run scripts

An efficient way to handle many simulations is to start them sequentially with the help of a script (a text file with commands); on computers with multi-core processors, it may be possible to start a few simulations at the same time, but when using a computer with

- single-core processor, or
- a single memory channel, or
- little memory,

running several simulations in parallel may actually increase the total computation time for the simulations.

Remarks:

See section Linux quick reference for useful Linux commands and examples on how to run and monitor FLACS-CFD commands effectively on a Linux system.

For example, the script file my_runfile could look like this on Linux:

```csh
#!/bin/csh -f
# Set up an alias for running the FLACS-CFD simulator:
alias my_runflacs /usr/local/Gexcon/FLACS-CFD_22.2/bin/run_runflacs
# Run the simulations in series:
my_runflacs 010101
my_runflacs 010102
my_runflacs 010103
Make the script file my.runfile executable:
> chmod u+x my_runfile
To run the script in the background, while sending output to the file my.listfile:
> ./my_runfile >& my_listfile &
Similarly, you can create a bat script, my.runfile.bat on Windows:
@echo off
setlocal
rem Set up an alias for running the FLACS-CFD simulator:
set my_runflacs="c:\program files\gexcon\flacs-cfd_22.2\bin\runflacs.exe"
rem Run the simulations in series:
%my_runflacs% 010100
%my_runflacs% 010101
%my_runflacs% 010102
and run it, writing output to the file my.listfile, by:
> my_runfile.bat > my_listfile

4.4.5 Stopping simulations from the command line

Flacs simulations can be stopped prematurely either by using the Task Manager (Windows) or using a combination of the command line utilities ps and kill on Linux.

To find the Flacs simulation's process ID (PID), run the following command on Linux:

> ps -edlaf | grep flacs
4.5 Different variants of the simulator

This will list all FLACS-CFD related programs running on the computer. The PID is the number found in the 4th column. Also note the user name of the process, which is found in the 3rd column. The process can then be stopped using the following command:

```
> kill 1234
```

where 1234 is the PID. Alternatively the simulation can be stopped by using the Runtime simulation control file (cc-file). Adding the following line to the cc-file will stop the simulation at the time 123.4 sec:

```
TSTOP 123.4
```

4.4.6 Optimising computer loads

In general Flacs simulations will use 100% of a single CPU core if available, thus optimal use of a computer for running simulations is to start as many simulations in parallel as there are CPU cores available. It is, however, necessary to keep the total memory consumption within the physical computer memory. If this is not done, the computer will start to use virtual memory (on the hard-disk), which is significantly slower than physical memory. This will result in longer simulation times.

As a rule of thumb, a computer for running Flacs simulations should have 2GB of memory per CPU core.

4.5 Different variants of the simulator

The following simulator variants are available in the FLACS-CFD installation:

- flacscfd handles the following simulation types:
  - (standard FLACS-CFD) - Dispersion and ventilation (requires a Dispersion or a Single Component license)
  - (standard FLACS-CFD) - Gas explosion (requires a GasEx or a Single Component license)
  - Pool (requires a Dispersion or a Single Component license)
  - Blast - explosives simulator (requires a Blast license)
  - Fire - fire simulator (requires a Fire or a Single Component license)
  - Dust explosions (requires a DustEx license)

It is possible to run the simulator executables directly, but we’d urge you to either use RunManager or runflacs when running simulations. This ensures that the correct simulator variant is chosen based on the settings in your scenario file.

Note:

The flacscfd simulator will only run simulations where the porosity file have been calculated using FGC (Flacs Geometry Calculator).

4.5.1 FLACS-CFD solver licensing

The FLACSCFD license feature was introduced in version 21.2 and is now used in place of the FLACS feature to decrypt the solver executable. Launching any simulation now requires the new FLACSCFD license feature.

A new per-thread licensing model, based on the FLACSCFD license features, was also introduced in version 21.2. The solver continues to support the old model, while also supporting this new model, and the old license model takes precedence if both are present. The following new license features are available in the new model:

- FLACSCFD Dispersion
• FLACSCFD GasEx
• FLACSCFD DustEx
• FLACSCFD Fire
• FLACSCFD Blast
• FLACSCFD Single Component <Cas_ID>

The Single Component license is a special license feature that allows simulations of dispersion, fire and gas explosion in scenarios that involve only one gas component which is identified by <Cas_ID>. This license feature takes precedence over other features if found and applicable.

In the new licensing model, each execution thread of the solver performs one login to the license feature. Only one of the license features listed above is used at a time, and the number of logins to the respective license feature is 1 for a serial run and NumThreads for parallel runs. As a consequence, the user is required to have a respective number of license features corresponding to the simulation type. The rule of thumb is that all Gas Explosion, Fire, Pool or Dust Explosion scenarios require FLACSCFD Dispersion license to run unless immediately ignited. The Dispersion license is freed and the respective license feature is acquired at ignition time. There are two exceptions to this rule:

• FLACSCFD Single Component license, as described above, can be used in a valid scenario for dispersion, fire and gas explosion modelling.
• FLACSCFD Fire license allows up to 1 and 5 seconds of dispersion simulation just before the ignition in jet-fire and pool-fire scenarios.

4.5.2 FLACS-CFD parallel solver

The parallel FLACS-CFD solver can utilise several CPU cores of a shared memory system in order to speed up simulations. Simulation speed depends on both the specific scenario and the used hardware, but can be increased by a multiple in many cases.

Note:

The most efficient way to use FLACS-CFD is to run many serial simulations at the same time, as is done in most FLACS-CFD studies which typically require a large number of simulations to be finished simultaneously. Running a single FLACS-CFD simulation in parallel speeds up that particular simulation compared to serial processing, but, as all parallel processing, introduces additional system overhead compared to serial execution so that the speed-up does not match the number of processors used.

Note:

Differences in results are expected between the Serial and the OMP solver. This is due to small numerical differences accumulating over time, for each iteration the accumulated difference might become more visible. For most cases these differences will be very minor. However, for some edge cases (especially where the flow is highly unstable), larger differences may be observed.

The following hardware is required to be able to exploit the parallel FLACS-CFD solver:

• a computer with more than one CPU core in a shared memory configuration (most current desktop computers have several cores even on a single processor),
• the bandwidth between the CPU cores and main memory must be high,
• important system specifications:
  – HyperTransport Technology (AMD) or Quick Path Interconnect (Intel),
4.5 Different variants of the simulator

- large number of memory channels,
- high memory bandwidth; state of the art (2012) on Intel/AMD based systems is 6400 MT/s (mega-transactions per second).

An example of results from tests performed on a Dell M915 server with 12 AMD Opteron 6180 CPUs and a total 48 CPU cores is shown below. The test system has 16 memory channels and a typical speed-up factor of 8 is seen.

![Performance curves for a 128³ grid case](image)

Figure 4.6: Typical performance of the parallel solver for increasing number of processors.

For studies that require to complete one or a small number of simulations quickly, FLACS-CFD parallel can help by reducing simulation time, if suitable hardware is available. Running the parallel solver on inadequate hardware may lead to lower performance than obtained using the standard solver.

4.5.2.1 Running the parallel solver from the command line

To run the FLACS-CFD parallel solver it has to be started as follows:

```plaintext
> run runflacs version _omp <jobnumber> NumThreads=4
```

NumThreads is optional and sets the number of CPU cores used by FLACS-CFD. If not set, FLACS-CFD will use maximum number of threads available.

4.5.2.2 Running the parallel solver from the FLACS-CFD RunManager

In the FLACS-CFD Runmanager click the Parameters button. Next, check Parallel run and set the number of threads to be used by FLACS-CFD in the dialog.

To make parallel runs the default, adjust the settings in the RunManager preferences. Also the default number of threads can be set there. Note that the parallel solver will always be used when the Parallel run check-box is enabled, even if the number of threads is set to 1.
4.5.3 **FLACS-CFD incompressible solver**

**Attention:**

It is not recommended to use the incompressible solver for any simulations, as performance benefit is minimal and it has not been sufficiently validated.

The standard FLACS-CFD simulator solves the governing equations for compressible flow. This is both necessary and appropriate for describing phenomena such as gas and dust explosions. A dedicated solver assuming incompressible flow is also available. This reduces the number of equations to be solved. However, despite this reduction in equations, any speed up observed is typically minimal. The incompressible solver is also less well validated than the standard solver. It is therefore recommended that the incompressible solver is not generally used.

4.5.3.1 **Technical details for the incompressible solver**

This section describes the technical modifications applied to the solver to create an incompressible version of FLACS-CFD.

- Governing equations: Incompressible flows are isochoric, i.e. volume remains constant, and the ideal gas law therefore becomes $p_0 = \rho RT$. This indicates that density varies with temperature and composition and the governing equations must therefore account for density variations. Therefore, the governing equations remain unchanged, except that,

1. $p + p_0$ is replaced with $p_0$ in the equation of state,
2. the $Dp/dt$ term in the enthalpy equation is omitted.
4.5 Different variants of the simulator

• Numerical schemes: The terms in the general discretised transport equations are identical for compressible and incompressible flows. However, minor modifications are performed to attain higher performance for incompressible flows as follows.

1. The speed of sound \( c \) for incompressible flows is, by definition, infinite. Therefore, terms containing \( 1/c^2 \) in the system of equations for pressure correction are omitted.

2. Since the aim of using an incompressible solver is to increase the time step, a lower value for the pressure under-relaxation is required in the SIMPLE algorithm to obtain convergence. However, this under-relaxation parameter is application-specific. Therefore, an alternative pressure correction algorithm called SIMPLEC (SIMPLE consistent) (Van Doormal & Raithby, 1984) has been implemented. An advantage of the SIMPLEC algorithm is that no user-specified under-relaxation parameter is needed for the pressure.

• Boundary conditions:

1. On the WIND boundaries, density is set equal to the reference density for incompressible flows.
2. BERNOULLI, EULER and NOZZLE boundary conditions are modified for incompressible flows.
3. A constant pressure boundary condition has been implemented for handling outflows when using the incompressible solver.

4.5.3.2 Using the incompressible solver

To use the incompressible solver in FLACS-CFD, INCOMPRESSIBLE must be set in Simulation and output control. The incompressible solver can be used in most dispersion and ventilation scenarios but be aware of the following:

• The Mach number should normally be below approximately 0.3.

• The Mach number limitation will vary from case to case. In some dispersion simulations, good results can be obtained even for flows up to Mach number 0.5, but the accuracy depends on many factors, including:
  – leak rate and location,
  – time step,
  – Mach number.

Attention:
Run sensitivity tests to verify the simulation results.

4.5.3.3 Simulating leaks with the incompressible solver

Leaks are modelled in several ways in FLACS-CFD: various jet leak models, diffusive leaks, and evaporation from pools. During the implementation of the incompressible solver, the FLACS-CFD source code has been reviewed with respect to the influence of the pressure on density for leaks, and the real pressure has been substituted with the ambient pressure \( p_0 \) wherever the density of leaking material is calculated. It should be noted that isentropic expansion from a pressurised vessel or infinitely large storage is assumed in the jet utility program. Provided the Mach number in the jet output is low enough, it is considered unproblematic to use the data computed for compressible fluids in the incompressible solver.
4.5.3.4 Time step for the incompressible solver

Time steps in transient simulations should be set such that the solution evolves smoothly and stably in time. The Courant-Friedrichs-Levy (CFL) number is used to give a solver-specific criterion for the maximum time step that gives a stable solution. In the compressible version of FLACS-CFD, two CFL numbers are used to determine the possible time step

\[
\Delta t_v = \frac{\text{CFLV}}{\max \left( \frac{u_i}{\Delta x_i} \right)} \quad \text{and} \quad \Delta t_c = \frac{\text{CFLC}}{\max \left( \frac{c}{\Delta x_i} \right)},
\]

(4.1)

for the advective transport and the propagation of sound waves, respectively. The length of the next time step is then taken as

\[
\Delta t = \min(\Delta t_v, \Delta t_c).
\]

(4.2)

The default values for explosion simulations are CFLV = 0.5 and CFLC = 5, while CFLV = 1.0 and CFLC = 10 are commonly used for dispersion simulations. For low velocities, the time step is limited by the speed of sound (\( \Delta t_c \)). In incompressible flow, the speed of sound criterion does not apply, which allows much longer time steps for simulations with low fluid velocities (assuming incompressibility implies that the speed of sound is infinite). However, disturbances are also transported by diffusion and a time step criterion based on the effective viscosity must be taken into account to ensure a stable solution for low-velocity cases. Hence, the time step in the incompressible flow solver is limited for advection and diffusion by

\[
\Delta t_v = \frac{\text{CFLV}}{\max \left( \frac{u_i}{\Delta x_i} \right)} \quad \text{and} \quad \Delta t_d = \frac{\text{CFLD}}{\max \left( \frac{\nu}{\rho \Delta x_i^2} \right)},
\]

(4.3)

respectively, and

\[
\Delta t = \min(\Delta t_v, \Delta t_d).
\]

(4.4)

For the diffusion criterion, CFLD = 2.0 is set in FLACS-CFD. Flow conditions like strong velocity gradients can lead to \( \Delta t_d \) being the limiting time step, so that increasing CFLV has no effect in those cases. For the incompressible solver, the CFLV number should be set as for the compressible solver to ensure stability and accuracy (see CFLV). It should be noted that while stability normally limits the time step for the compressible solver, the time step for the incompressible solver may be limited by accuracy considerations. In other words, increasing the CFLV number (or directly increasing the time step) may yield enhanced performance without stability problems, but the solution accuracy may deteriorate significantly as the time-step becomes longer.

In quiescent conditions with wind build-up, the maximum flow velocity may be negligibly small before the onset of the wind (or other phenomena). To avoid very long time steps, it is therefore recommended to set a minimum velocity to be used for determining the time step via the CFLV criterion. This can be achieved by setting STEP "V_MIN=5.0" in the SIMULATION_AND_OUTPUT_CONTROL section of the cs-file or by using the V_MIN in CASD. The value, measured in m/s, is used as the flow velocity for determining the time step via the CFLV criterion whenever the actual maximum flow velocity is lower. It is generally appropriate to set this value to equal the inflow velocity after the wind build-up time, or to a typical macroscopic flow velocity for the scenario.

4.5.4 flacscfd

flacscfd is the next-generation simulator that replaces most of the (cf. Flacs2 variants) in the current releases of the FLACS-CFD package. The flacscfd simulator can be started from The RunManager or from the command line using the steps described in Starting simulations. flacscfd exists in both serial and parallel versions, see Running the parallel from the command line for more information about how the different versions can be run.

Key features of flacscfd are described in the subsequent sections.
4.5 Different variants of the simulator

4.5.4.1 Steady-state solver

A steady-state can be described as a situation in which all state variables are constant in spite of ongoing processes that strive to change them. A steady-state is usually approached asymptotically, and true steady-state can be reached as time approaches infinity. In practical applications, steady-state can be defined as a state where the rate of change of state variables fit within predefined tolerances, and is sometimes referred to as a convergence to a steady-state.

A simple example of a system that have reached steady-state is a fixed rate jet leak. As time approaches infinity, the fuel flows in and out of the domain at the same rate. If all other state variables (e.g. pressure and velocity field) also stabilise, the system is considered to be in a steady-state.

FLACS-CFD solves the transient version of the governing equations for compressible flow. The steady-state solver is a collection of modifications of numerical and control parameters applied to the solver that allow achieving a stable and fast convergence to a steady-state.

The current implementation of the steady-state solver uses large time steps of global pseudo-time to reach a convergence. The solution is stabilised by the use of reduced underrelaxation factors (see Stabilising solution for more details). As a result, in most cases, a much faster convergence to a steady-state solution is achieved. The order of speedup varies between 5 and 50 in practical applications. The consequence of using pseudo-time instead of physical time is that time is no longer physically meaningful. This has the following implications:

- The iteration log is changed.
- Ignition is disabled.
- Intermediate results are not written to the output file.
- The CFL numbers set in Simulation and output control are ignored.
- The LAST parameter, set in Simulation and output control, indicates the maximum number of iterations allowed.
- A constant leak rate is assumed, and the leak files are regenerated from leaks in the Simulation and output control section. The START TIME and DURATION parameters of the leaks are ignored.

The steady-state solver can be enabled by specifying STEADY in the Solver mode (MODE) mode field in the Simulation and output control section of the scenario.

4.5.4.1.1 Limitations  The steady-state solver have a number of limitations that are typical for this class of solvers:

- The nature of the problem must be steady. The steady-state solver cannot be applied to problems that are transient by nature because it will never converge (time-dependent release, explosions, blast propagation, vortex shedding).
- All intermediate results have no physical interpretation and is therefore excluded from the simulation output.
- The iteration algorithm have been extensively tested and have demonstrated good performance, however, stability issues might appear for low density and high velocity flows. See the Stabilising solution section for how to handle instability and divergence issues.
- No physical models have been changed in the steady-state solver; therefore, as long as the solution converges, the results should be similar to the transient solver.

Note:

Results of the simulation are written to the CGNS file on normal exit only - when simulation converges, any of the stop criteria is met or simulation is suspended. Aborted simulations will not write final results. It is advised to always interrupt simulations using suspend functionality.
4.5.4.1.2 Stabilising solution  Among many parameters that can affect performance and stability (but not the accuracy) of the solution, underrelaxation factors \texttt{UNDERRELAX}_P and \texttt{UNDERRELAX}_{UVW} were found to be the most effective. Their values can be specified in the Simulation and output control section of the scenario.

In case of convergence or stability problems, the user is encouraged to try the following:

- reduce value of \texttt{UNDERRELAX}_P starting with half of the default value;
- if the above doesn’t help, reduce also value of \texttt{UNDERRELAX}_{UVW} starting with half of the default value;
- if the above doesn’t help, keep reducing both values further;
- \(1/8\)th of the original values is the practical limit where it is better to consider running full transient simulation instead of trying to stabilize steady solution.

Note:
Lower underrelaxation factors stabilize the solution, while higher values lead to a quicker convergence. Residual values for corresponding equations are non-linearly correlated with the value of the underrelaxation factor. Therefore it is challenging to establish one set of the convergence criteria that would work for all cases in the wide range of the underrelaxation factor values. In order to make the steady-state solver more reliable, the automatic convergence checking algorithm (\texttt{CHECK\_CONVERGENCE}) in FLACS-CFD detects whether the underrelaxation factors are set to the non-default values and keeps adjusting their values until the convergence for the default values is reached.

4.6 Output variables in FLACS-CFD

This chapter describes the output variables that can be selected for output in selected monitor points or over selected monitor panels (under 'Single Field Scalar Time Output' on the Scenario menu in CASD), or throughout the entire calculation domain (under 'Single Field 3D Output' on the Scenario menu in CASD). Each variable name in \texttt{flascfd} has additionally a SIDS data-name identifier defined. These standardized names are used to access data in CGNS files through Python API and in Flowvis.

The normal output variables in FLACS-CFD are:

<table>
<thead>
<tr>
<th>Name</th>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UVW</td>
<td>VelocityMagnitude</td>
<td>(m/s)</td>
<td>Velocity value</td>
</tr>
<tr>
<td>VVEC</td>
<td>VelocityVector</td>
<td>(m/s)</td>
<td>Velocity vector (dim=3)</td>
</tr>
<tr>
<td>U</td>
<td>VelocityX</td>
<td>(m/s)</td>
<td>Velocity component x-direction</td>
</tr>
<tr>
<td>V</td>
<td>VelocityY</td>
<td>(m/s)</td>
<td>Velocity component y-direction</td>
</tr>
<tr>
<td>W</td>
<td>VelocityZ</td>
<td>(m/s)</td>
<td>Velocity component z-direction</td>
</tr>
<tr>
<td>H</td>
<td>Enthalpy</td>
<td>(J/kg)</td>
<td>Enthalpy</td>
</tr>
<tr>
<td>FUEL</td>
<td>MassFractionFuel</td>
<td>(-)</td>
<td>Fuel mass fraction</td>
</tr>
<tr>
<td>FUEL_MAX</td>
<td>MassFractionFuel_Max</td>
<td>(-)</td>
<td>Maximum fuel mass fraction</td>
</tr>
<tr>
<td>FMIX</td>
<td>MassFractionMixture</td>
<td>(-)</td>
<td>Mixture fraction</td>
</tr>
<tr>
<td>FMIX_MAX</td>
<td>MassFractionMixture_Max</td>
<td>(-)</td>
<td>Maximum mixture fraction</td>
</tr>
<tr>
<td>FVAR</td>
<td>ExpansionRatio</td>
<td>(-)</td>
<td>Expansion ratio</td>
</tr>
<tr>
<td>K</td>
<td>TurbulentEnergyKinetic</td>
<td>(m2/s2)</td>
<td>Turbulent kinetic energy</td>
</tr>
<tr>
<td>EPK</td>
<td>TurbulentDissipationRate</td>
<td>(1/s)</td>
<td>Turbulent dissipation rate / Specific dissipation rate of turbulent kinetic energy</td>
</tr>
</tbody>
</table>

FLACS-CFD v22.2 User’s Manual
## 4.6 Output variables in FLACS-CFD

<table>
<thead>
<tr>
<th>Name</th>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EPS</td>
<td>TurbulentDissipation</td>
<td>(m²/s³)</td>
<td>Dissipation rate of turbulent kinetic energy</td>
</tr>
<tr>
<td>GAMMA</td>
<td>SpecificHeatRatio</td>
<td>(-)</td>
<td>Ratio of specific heats / Isentropic gas constant</td>
</tr>
<tr>
<td>LT</td>
<td>TurbulentLengthScale</td>
<td>(m)</td>
<td>Turbulent length scale</td>
</tr>
<tr>
<td>MU</td>
<td>ViscosityEffective</td>
<td>(kg/(m*s))</td>
<td>Effective dynamic viscosity</td>
</tr>
<tr>
<td>OX</td>
<td>MassFractionOxygen</td>
<td>(-)</td>
<td>Oxygen mass fraction</td>
</tr>
<tr>
<td>P</td>
<td>Pressure</td>
<td>(barg)</td>
<td>Pressure</td>
</tr>
<tr>
<td>PMAX</td>
<td>PressureMax</td>
<td>(barg)</td>
<td>Maximum pressure</td>
</tr>
<tr>
<td>PMIN</td>
<td>PressureMin</td>
<td>(barg)</td>
<td>Minimum pressure</td>
</tr>
<tr>
<td>DPDT</td>
<td>PressureRiseRate</td>
<td>(bar/s)</td>
<td>Rate of pressure rise</td>
</tr>
<tr>
<td>PIMP</td>
<td>PressureImpulse</td>
<td>(Pa*s)</td>
<td>Pressure impulse</td>
</tr>
<tr>
<td>PIMP_MAX</td>
<td>PressureImpulseMax</td>
<td>(Pa*s)</td>
<td>Maximum pressure impulse</td>
</tr>
<tr>
<td>PROD</td>
<td>MassFractionProducts</td>
<td>(-)</td>
<td>Combustion product mass fraction</td>
</tr>
<tr>
<td>PROD_MAX</td>
<td>MassFractionProducts_Max</td>
<td>(-)</td>
<td>Maximum combustion product mass fraction</td>
</tr>
<tr>
<td>PTIMP</td>
<td>PressureStagnationImpulse</td>
<td>(Pa*s)</td>
<td>Total(stagnation) pressure impulse</td>
</tr>
<tr>
<td>PTOT</td>
<td>PressureStagnation</td>
<td>(Pa)</td>
<td>Total(stagnation) pressure</td>
</tr>
<tr>
<td>RFU</td>
<td>CombustionRate</td>
<td>(kg/(m³*s))</td>
<td>Combustion Rate</td>
</tr>
<tr>
<td>RET</td>
<td>ReynoldsTurbulent</td>
<td>(-)</td>
<td>Turbulent Reynolds number</td>
</tr>
<tr>
<td>FMOLE</td>
<td>MoleFractionFuel</td>
<td>(m³/m³)</td>
<td>Fuel mole fraction</td>
</tr>
<tr>
<td>FMOLE_MAX</td>
<td>MoleFractionFuel_Max</td>
<td>(m³/m³)</td>
<td>Maximum fuel mole fraction</td>
</tr>
<tr>
<td>FDOSE</td>
<td>DoseFuel</td>
<td>(m³/m³*s)</td>
<td>Fuel mole fraction DOSE</td>
</tr>
<tr>
<td>RHO</td>
<td>Density</td>
<td>(kg/m³)</td>
<td>Density</td>
</tr>
<tr>
<td>T</td>
<td>Temperature</td>
<td>(K)</td>
<td>Temperature</td>
</tr>
<tr>
<td>TURB</td>
<td>TurbulenceVelocity</td>
<td>(m/s)</td>
<td>Turbulence velocity</td>
</tr>
<tr>
<td>TURBI</td>
<td>TurbulenceIntensityRel</td>
<td>(-)</td>
<td>Relative turbulence intensity</td>
</tr>
<tr>
<td>UDRAG</td>
<td>DragX</td>
<td>(Pa)</td>
<td>Drag component x-direction</td>
</tr>
<tr>
<td>VDRAG</td>
<td>DragY</td>
<td>(Pa)</td>
<td>Drag component y-direction</td>
</tr>
<tr>
<td>WDRAG</td>
<td>DragZ</td>
<td>(Pa)</td>
<td>Drag component z-direction</td>
</tr>
<tr>
<td>DRAG</td>
<td>DragMagnitude</td>
<td>(Pa)</td>
<td>Drag value</td>
</tr>
<tr>
<td>DRAGMAX</td>
<td>DragMagnitudeMax</td>
<td>(Pa)</td>
<td>Maximum drag value</td>
</tr>
<tr>
<td>UDIMP</td>
<td>DragXImpulse</td>
<td>(Pa*s)</td>
<td>Drag-impulse component x-direction</td>
</tr>
<tr>
<td>VDIMP</td>
<td>DragYImpulse</td>
<td>(Pa*s)</td>
<td>Drag-impulse component y-direction</td>
</tr>
<tr>
<td>WDIMP</td>
<td>DragZImpulse</td>
<td>(Pa*s)</td>
<td>Drag-impulse component z-direction</td>
</tr>
<tr>
<td>DIMP</td>
<td>DragImpulse</td>
<td>(Pa*s)</td>
<td>Drag-impulse value</td>
</tr>
<tr>
<td>DIMP_MAX</td>
<td>DragImpulseMax</td>
<td>(Pa*s)</td>
<td>Maximum drag-impulse value</td>
</tr>
<tr>
<td>UMACH</td>
<td>MachX</td>
<td>(-)</td>
<td>Mach number component x-direction</td>
</tr>
<tr>
<td>VMACH</td>
<td>MachY</td>
<td>(-)</td>
<td>Mach number component y-direction</td>
</tr>
</tbody>
</table>
### Table 4.2: Scalar (dim=1) field (3D) output variables in FLACS-Fire

<table>
<thead>
<tr>
<th>Name</th>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMACH</td>
<td>MachZ</td>
<td>(-)</td>
<td>Mach number component z-direction</td>
</tr>
<tr>
<td>MACH</td>
<td>Mach</td>
<td>(-)</td>
<td>Mach number value</td>
</tr>
<tr>
<td>CS</td>
<td>VelocitySound</td>
<td>(m/s)</td>
<td>Sound velocity</td>
</tr>
<tr>
<td>ER</td>
<td>EquivalenceRatio</td>
<td>(-)</td>
<td>Equivalence ratio</td>
</tr>
<tr>
<td>ER_MAX</td>
<td>EquivalenceRatio_MAX</td>
<td>(-)</td>
<td>Maximum equivalence ratio</td>
</tr>
<tr>
<td>ERLFL</td>
<td>EquivalenceRatioLFL</td>
<td>(-)</td>
<td>Equivalence ratio as %LFL</td>
</tr>
<tr>
<td>ERLFL_MAX</td>
<td>EquivalenceRatioLFL_Max</td>
<td>(-)</td>
<td>Maximum equivalence ratio as %LFL</td>
</tr>
<tr>
<td>ERNFL</td>
<td>EquivalenceRatioNFL</td>
<td>(-)</td>
<td>Equivalence ratio, normalised flammable range</td>
</tr>
<tr>
<td>EQ</td>
<td>EquivalenceRatioFinBnd</td>
<td>(-)</td>
<td>Equivalence ratio, finite bounded</td>
</tr>
<tr>
<td>EQLFL</td>
<td>EquivalenceRatioFinBndLFL</td>
<td>(-)</td>
<td>Equivalence ratio, finite bounded, as %LFL</td>
</tr>
<tr>
<td>EQNFL</td>
<td>EquivalenceRatioFinBndNFL</td>
<td>(-)</td>
<td>Equivalence ratio, finite bounded, normalised flammable range</td>
</tr>
<tr>
<td>TMOLE</td>
<td>ToxicMoleFraction</td>
<td>(m3/m3)</td>
<td>Toxic mole fraction</td>
</tr>
<tr>
<td>TCONS</td>
<td>ToxicConcentration</td>
<td>(mg/m3)</td>
<td>Toxic concentration</td>
</tr>
<tr>
<td>TDOSE</td>
<td>ToxicDose</td>
<td>(mg/m3+minute)</td>
<td>Toxic dose</td>
</tr>
<tr>
<td>PROBIT</td>
<td>ToxicProbit</td>
<td>(-)</td>
<td>Toxic probit</td>
</tr>
<tr>
<td>PDEATH</td>
<td>ToxicDeathProbability</td>
<td>(-)</td>
<td>Probability of death as function of toxic probit</td>
</tr>
<tr>
<td>DPDX</td>
<td>PressureGradientNormalized</td>
<td>(-)</td>
<td>Normalized spatial pressure gradient</td>
</tr>
<tr>
<td>DDTLS</td>
<td>DetonationLengthScaleRatio</td>
<td>(-)</td>
<td>Detonation length scale ratio</td>
</tr>
<tr>
<td>POOL_D</td>
<td>PoolDepth</td>
<td>(m)</td>
<td>Pool depth</td>
</tr>
<tr>
<td>POOL_VOL</td>
<td>PoolVolumeFillFraction</td>
<td>(-)</td>
<td>Pool volume fill fraction (derived from POOL_D)</td>
</tr>
<tr>
<td>POOL_T</td>
<td>PoolTemperature</td>
<td>(K)</td>
<td>Pool temperature</td>
</tr>
<tr>
<td>POOL_U</td>
<td>PoolVelocityX</td>
<td>(m/s)</td>
<td>Pool velocity component in x-direction</td>
</tr>
<tr>
<td>POOL_V</td>
<td>PoolVelocityY</td>
<td>(m/s)</td>
<td>Pool velocity component in y-direction</td>
</tr>
<tr>
<td>POOL_S</td>
<td>PoolVelocityVector</td>
<td>(m/s)</td>
<td>Pool velocity vector (dim=2)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSCOF</td>
<td>RadAbsCoeff</td>
<td>(1/m)</td>
<td>Absorption coefficient</td>
</tr>
<tr>
<td>HCONV</td>
<td>ConvHTCoeff</td>
<td>(W/m2/K)</td>
<td>Convective heat transfer coefficient</td>
</tr>
<tr>
<td>RADSRC</td>
<td>RadSourceTerm</td>
<td>(kW/m3)</td>
<td>Radiation source</td>
</tr>
<tr>
<td>Q</td>
<td>TotHeatFlux</td>
<td>(kW/m2)</td>
<td>Total heat flux</td>
</tr>
<tr>
<td>QCONV</td>
<td>ConvHeatFlux</td>
<td>(kW/m2)</td>
<td>Convective heat flux</td>
</tr>
<tr>
<td>QRAD</td>
<td>RadHeatFlux</td>
<td>(kW/m2)</td>
<td>Radiative heat flux</td>
</tr>
<tr>
<td>QRADFF</td>
<td>RadFarFieldHeatFlux</td>
<td>(kW/m2)</td>
<td>Far-field radiative heat flux</td>
</tr>
</tbody>
</table>
4.6 Output variables in FLACS-CFD

<table>
<thead>
<tr>
<th>Name</th>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QRINC</td>
<td>RadIncHeatFlux</td>
<td>(kW/m²)</td>
<td>Incident radiative flux</td>
</tr>
<tr>
<td>QWALL / QTINC</td>
<td>TotIncHeatFlux</td>
<td>(kW/m²)</td>
<td>Total incident heat flux</td>
</tr>
<tr>
<td>QDOSE</td>
<td>TotHeatDose</td>
<td>((kW/m²)^(4/3)s)</td>
<td>Heat dose</td>
</tr>
<tr>
<td>TWALL</td>
<td>WallTemperature</td>
<td>(K)</td>
<td>Wall temperature</td>
</tr>
<tr>
<td>CO₂</td>
<td>MassFractionCO₂</td>
<td>(-)</td>
<td>Mass fraction of CO₂</td>
</tr>
<tr>
<td>H₂O</td>
<td>MassFractionH₂O</td>
<td>(-)</td>
<td>Mass fraction of H₂O</td>
</tr>
<tr>
<td>SOOT</td>
<td>MassFractionSoot</td>
<td>(-)</td>
<td>Mass fraction of soot</td>
</tr>
<tr>
<td>VFCO₂</td>
<td>VolumeFractionCO₂</td>
<td>(-)</td>
<td>Volume fraction of CO₂</td>
</tr>
<tr>
<td>VFH₂O</td>
<td>VolumeFractionH₂O</td>
<td>(-)</td>
<td>Volume fraction of H₂O</td>
</tr>
<tr>
<td>VFSOOT</td>
<td>VolumeFractionSoot</td>
<td>(-)</td>
<td>Volume fraction of soot</td>
</tr>
<tr>
<td>VISIB</td>
<td>Visibility</td>
<td>(m)</td>
<td>Visibility</td>
</tr>
</tbody>
</table>

Table 4.3: Scalar (dim=1) field (3D) output variables in DustEx

<table>
<thead>
<tr>
<th>Name</th>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DUST_C</td>
<td>DustConcentration</td>
<td>(-)</td>
<td>Dust concentration</td>
</tr>
<tr>
<td>DUST_B</td>
<td>DustConcentrationBurnable</td>
<td>(-)</td>
<td>Dust burnable concentration</td>
</tr>
<tr>
<td>STOKES</td>
<td>StokesNumber</td>
<td>(-)</td>
<td>Dust Stokes number</td>
</tr>
<tr>
<td>INERT_C</td>
<td>InertConcentration</td>
<td>(-)</td>
<td>Concentration of inerts</td>
</tr>
</tbody>
</table>

The panel output variables in FLACS-CFD are:

<table>
<thead>
<tr>
<th>Name</th>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDIMP</td>
<td>PanelDragImpulse</td>
<td>(Pa·s)</td>
<td>Panel average drag impulse</td>
</tr>
<tr>
<td>PDRAG</td>
<td>PanelDrag</td>
<td>(Pa)</td>
<td>Panel average drag</td>
</tr>
<tr>
<td>PP</td>
<td>PanelPressure</td>
<td>(Pa)</td>
<td>Panel average pressure</td>
</tr>
<tr>
<td>PPIMP</td>
<td>PanelPressureImpulse</td>
<td>(Pa·s)</td>
<td>Panel average pressure impulse</td>
</tr>
<tr>
<td>PPOR</td>
<td>PanelPorosity</td>
<td>(-)</td>
<td>Panel average area porosity</td>
</tr>
<tr>
<td>PTRA</td>
<td>PanelTraveledDistance / PanelRotationAngle</td>
<td>(m) / (deg)</td>
<td>Panel traveled distance (POPOUT RIGID) / panel rotation angle (HINGED RIGID)</td>
</tr>
</tbody>
</table>

The panel output variables in FLACS-Fire are:

<table>
<thead>
<tr>
<th>Name</th>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PT</td>
<td>PanelAvgTemp</td>
<td>(K)</td>
<td>Panel average temperature</td>
</tr>
<tr>
<td>PQFFDOSE</td>
<td>PanelAvgDose</td>
<td>((kW/m²)^(4/3)s)</td>
<td>Panel average far-field radiation dose</td>
</tr>
<tr>
<td>PQRADFF</td>
<td>PanelRadFlux</td>
<td>(kW/m²)</td>
<td>Panel average far-field radiative flux</td>
</tr>
<tr>
<td>PQ</td>
<td>PanelTotFlux</td>
<td>(kW/m²)</td>
<td>Panel average total heat flux</td>
</tr>
<tr>
<td>PQTINC</td>
<td>PanelAvgTotInciFlux</td>
<td>(kW/m²)</td>
<td>Panel average total incident heat flux</td>
</tr>
</tbody>
</table>
The dimension (dim) of the output variables refers to how many values are written for each cell. Its value is needed only when editing the scenario file directly.

Attention:

Note that the MAX and IMP variables are reset when using a dump file. So are Q6 and Q7 output.

The following sections give a description of the most commonly used output variables.

4.6.1 General 3D output variables

Note that some of the variables presented in the following are also available as monitor point output. Having a scenario open in CASD, it is easy to check which variables are available for monitor point or 3D output by checking the scenario menu sections “Monitor points” and “Single field 3D output”.

4.6.1.1 Mass fraction of fuel: FUEL

This is the mass fraction of fuel in the mixture of fuel, air, and combustion products. The fuel may be a mixture of several components, such as hydrocarbons and hydrogen. Plots of FUEL are useful for displaying the fuel cloud.

4.6.1.2 Pressure: P

This is the static overpressure (bar g). In a flowing fluid, the total pressure is the sum of the static pressure and the dynamic pressure. The static pressure is isotropic, whereas the dynamic pressure, caused by the relative motion of the fluid, is anisotropic. A pressure transducer placed in a flow field will in general measure the static pressure and a certain portion of the dynamic pressure, depending on the orientation of the face of the pressure transducer relative to the flow direction. 'Head on' measurements give the total pressure, whereas 'side on' measurements give the static pressure.

4.6.1.3 Pressure impulse: P_IMP

P_IMP is the time integral of the pressure:

\[ I_p = \int_{t_1}^{t_2} pdt \]  

(4.5)

The pressure impulse is simply the area below the pressure-time curve, and since it is the product of pressure and time it holds information about both the amplitude and the duration of the pressure-time curve.

4.6.1.4 Mass fraction of combustion products: PROD

This is the ratio of mass (kg) of combustion products per unit mass (1 kg) of the total mixture of fuel, air and combustion products for each control volume. The combustion products consist of carbon-dioxide and water vapour (and also sulphur dioxide (SO2) when hydrogen sulfide (H2S) is included in the fuel composition). Note that nitrogen as part of air, is not included in the combustion products (Air is modelled as a mixture of oxygen and nitrogen). When, for example, a stoichiometric methane-air mixture has completely combusted, then the mass fraction PROD plus the mass fraction of nitrogen, equals 1.0. Plots of PROD are useful for displaying the flame (or more correctly the burnt volume).

See sections Definitions and gas thermodynamics, Stoichiometric reaction and Gas composition and volume for more information on the reactions that convert fuel onto combustion products.
4.6 Output variables in FLACS-CFD

4.6.1.5 Gas density: RHO

The gas density is:

\[ p = \rho RT \]  

(4.6)

This is the fluid mass (kg) per unit volume (1 m3). The equation of state gives the relation between pressure density and temperature.

4.6.1.6 Gas temperature: T

This is the absolute temperature (K) of the fluid. See RHO above for a description of the relation between pressure density and temperature. The temperature may be increased by compression which converts mechanical energy into thermal energy, and by combustion which converts chemical energy into thermal energy.

4.6.1.7 Turbulence velocity: TURB

The turbulent kinetic energy per unit of mass \( K \) [m²/s²] is related to the turbulence velocity components \( \text{TURB}_U \) [m/s], \( \text{TURB}_V \) [m/s], \( \text{TURB}_W \) [m/s], by the formula:

\[ K = \frac{1}{2} (\text{TURB}_U^2 + \text{TURB}_V^2 + \text{TURB}_W^2) \]  

(4.7)

When assuming isotropic turbulence so that

\[ \text{TURB} = \text{TURB}_U = \text{TURB}_V = \text{TURB}_W \]  

(4.8)

then a simplified formula for the turbulent kinetic energy per unit of mass \( K \) [m²/s²] becomes:

\[ K = \frac{3}{2} \text{TURB}^2 \]  

(4.9)

And rearranging this formula, an expression for the turbulence velocity \( \text{TURB} \) [m/s] becomes:

\[ \text{TURB} = \sqrt{\frac{2}{3} K} \]  

(4.10)

4.6.1.8 Relative turbulence intensity: TURBI

The output for the relative turbulence intensity, \( \text{TURBI} \), depends on the initial and boundary conditions. When the characteristic velocity (\text{CHARACTERISTIC_VELOCITY}) in the \text{INITIAL_CONDITIONS} section in the scenario file is

- zero, and the \text{WIND} boundary condition is applied for a positive wind speed, then \( \text{TURBI} \) [-] equals \( \text{TURB} \) [m/s] divided by the wind speed [m/s]. If, for example, \text{WIND_SPEED} 10, then \( \text{TURBI} \) equals \( \text{TURB} \) divided by 10;

- a positive number greater than the absolute value of the wind speed (if a \text{WIND} boundary condition is applied at all), then \( \text{TURBI} \) [-] equals \( \text{TURB} \) [m/s] divided by the characteristic velocity [m/s]. If, for example \text{CHARACTERISTIC_VELOCITY} 12 and \text{WIND_SPEED} 10, then \( \text{TURBI} \) equals \( \text{TURB} \) divided by 12;

- zero and the \text{WIND} boundary condition is \textit{not} applied, then \( \text{TURBI} \) [-] equals \( \text{TURB} \) [m/s] divided by the local value of \( \text{UVW} \) [m/s].
4.6.1.9 Velocity vector: VVEC
This is the entity which gives the three velocity components of the time averaged fluid flow. The energy contained in the temporal fluctuations of the flow which are not captured using a given spatial and temporal resolution is handled by a turbulence model. VVEC consists of the three components \( U, V, \) and \( W \). If you are editing the cs-file manually, always remember to include the components if you have specified VVEC for output (CASD includes them automatically).

4.6.1.10 Drag value: DRAG
The drag value (in FLACS-CFD defined as drag force per unit area) is proportional to the dynamic pressure for the fluid flow. The expression for the dynamic pressure is:

\[
p_{\text{dyn}} = \rho u^2 / 2
\]

(4.11)

An obstacle submerged in a fluid flow will interact with the fluid, thereby a drag force results. The drag force may be measured in experiments and if the Reynolds number is high, the ratio 'drag force / dynamic pressure' is constant:

Drag coefficient \( C_D = (F_D / A) / (\rho u^2 / 2) \)

Drag force \( F_D = C_D (\rho u^2 / 2) A \)

The drag value is calculated by assuming the drag coefficient \( C_D = 1 \) and the cross-section area \( A = 1 \), with this definition the drag value is the same as the dynamic pressure.

4.6.1.11 Drag-impulse value: DRAG_IMP
Drag-impulse value is the time integral of the dynamic pressure:

\[
I_{p_{\text{dyn}}} = \int_{t_1}^{t_2} p_{\text{dyn}} dt
\]

(4.12)

The drag-impulse value is equivalent to the pressure impulse, with the difference that the dynamic pressure is being integrated instead of the static pressure.

4.6.1.12 Equivalence ratio: ER
The equivalence ratio, \( ER \), is a measure of the concentration of fuel compared to the stoichiometric concentration, i.e. \( ER \) equals unity at stoichiometric concentration. If \( (F/O) \) is the ratio of fuel to oxygen, the equivalence ratio is defined as follows:

\[
ER = (F/O) / (F/O)_{\text{stoichiometric}}
\]

(4.13)

For zero fuel, \( ER \) equals zero and for pure fuel \( ER \) goes to infinity.

4.6.1.13 Equivalence ratio as %LFL: ERLFL
The output parameter ERLFL denotes a fuel concentration measure and is defined as

\[
ERLFL = 100 \times ER / ER_{LFL}[\%]
\]

(4.14)

That is, ERLFL compares the local fuel concentration to the LFL (lower flammable limit) concentration, and gives a percentage value.

The LFL value normally varies with gas type and oxygen concentration (again depending on the amount of inert gases) in the mixture. In FLACS-CFD the fuel is always mixed with air which has a preset oxygen concentration, so only the variation of LFL for the fuel type remains.

Note:

The name of this output parameter, ERLFL, is easy to confuse with the equivalence ratio at LFL, usually denoted \( ER_{LFL} \) in this manual. ERLFL depends on time and space, while \( ER_{LFL} \) is fixed for a given gas composition and state.
4.6 Output variables in FLACS-CFD

4.6.1.14 Equivalence ratio, normalised DFL: ERNFL

The flammable range is defined to be from LFL to UFL, where LFL is the lower flammable limit and UFL is the upper flammable limit. ERNFL is defined as follows:

\[
ERNFL = \frac{ER - ER_{LFL}}{ER_{UFL} - ER_{LFL}}
\]  

(4.15)

ERNFL is zero at LFL and one at UFL.

4.6.1.15 Equivalence ratio, finite bounded: EQ

This is a measure for concentration of fuel similar to the equivalence ratio (see ER above). Say that (F/O) is the ratio of fuel to oxygen, then the finite bounded equivalence ratio is defined as follows:

\[
EQ = \frac{F/O}{(F/O) + (F/O)_{stoichiometric}} = \frac{ER}{ER + 1}
\]

(4.16)

At stoichiometric concentration EQ equals 1/2. For zero fuel EQ equals zero and for pure fuel EQ equals one.

4.6.1.16 Equivalence ratio as %LFL: EQLFL

This is a measure for concentration of fuel compared to the LFL concentration, where LFL is the lower flammable limit. The LFL value normally varies with gas type and oxygen concentration (again depending on the amount of inert gases) in the mixture. In FLACS-CFD the fuel is always mixed with air which has a preset oxygen concentration, so only the variation of LFL with gas type remains. The definition of EQLFL is as follows:

\[
EQLFL = 100 \times \frac{EQ}{EQ_{LFL}} [%]
\]  

(4.17)

4.6.1.17 Equivalence ratio, normalised DFL: EQNFL

Equivalence ratio, normalised flammable range. The flammable range is defined to be from LFL to UFL, where LFL is the lower flammable limit and UFL is the upper flammable limit. EQNFL is defined as follows:

\[
EQNFL = \frac{EQ - EQ_{LFL}}{EQ_{UFL} - EQ_{LFL}}
\]  

(4.18)

EQNFL is zero at LFL and one at UFL.

4.6.1.18 FMOLE and FDOSE

FMOLE is the mole, or volume, fraction of the gas in the gas/air mixture, and FDOSE is the integrated (accumulated) FMOLE. For the 60s dose at the monitor points, you can export FMOLE to ASCII-format using r1file, import to excel, and subtract the FDOSE value of time-60s. The FDOSE variable can also be selected for 3D output. If you want a contour plot of the 60s exposure one or more plots must be selected for each 60s period. Using the r3file utility program one can then generate new r3-files with FDOSE(time)-FDOSE(time-60s).

The utility program r3file can generate the so-called dose (i.e. exposure) output:

\[
FDOSE(t) = \int_0^t FMOLE(t) dt
\]

dose(t2) = FDOSE(t2) - FDOSE(t1)

(dose/time)(t2) = dose(t2)/(t2 - t1)

The times (t2 and t1) are taken from the output times with a certain integer interval given by the option 'dose=' or 'dose/time='. If you want a contour plot of the 60s exposure one or more plots must be selected for each 60s period. Using the r3file utility program one can then generate new r3-files with FDOSE(time)-FDOSE(time-60s).

The times (t2 and t1) are taken from the output times with a certain integer interval given by the option 'dose=' or 'dose/time=':

\[
dose=2 \quad \text{means that t2-t1 = 2*DTPLLOT, output is then dose(t)}
\]

\[
dose/time=2 \quad \text{is similar, but you get (dose/time)(t) output}
\]
The following starting point is assumed:

- FLACS-CFD result files in the current directory, and
- FDOSE output at regular time intervals (e.g. DTPLT = 60)

To generate the dose (i.e. exposure) output in a separate directory:

1. create the directory 'work' and enter into it
2. run the r3file utility (assuming the job number is 010100)

```plaintext
> mkdir work
> cd work
> run r3file1.3 ../r3010100.dat3 format=r3file dose=1 name=NFDOSE force
```

Now the following files can be found in the working directory:

```
a3010100.NFDOSE
cgNFDOSE.dat3 -> ../cg010100.dat3
c0NFDOSE.dat3 -> ../co010100.dat3
cpNFDOSE.dat3 -> ../cp010100.dat3
csNFDOSE.dat3
cpNFDOSE.dat3 -> a3010100.NFDOSE
```

The results can be examined using Flowvis, choose the result file with the name NFDOSE instead of a job number in the Select files dialog box.

**Warning:**

'region=' cannot be used together with the dose output.

![Plotting of NFDOSE in Flowvis](image)

**4.6.2 3D output variables for fire simulations**

The following output variables are available for simulations using FLACS-Fire.

**4.6.2.1 Wall temperature: TWALL**

By default the wall temperature is a constant (in time and space) and equal to the input value. To dynamically calculate the wall temperature for a given surface, the conduction model needs to be activated. TWALL is calculated at wall surfaces only.
4.6.2.2 Mass fractions: SOOT, H2O, CO2

The variables SOOT, H2O and CO2 are mass fractions of \( \text{soot} \), \( H_2O \) and \( CO_2 \), respectively.

4.6.2.3 Volume/mole fractions: VFSOOT, VFH2O, VFCO2

The variables VFSOOT, VFH2O and VFCO2 are volume/mole fractions of \( \text{soot} \), \( H_2O \) and \( CO_2 \) respectively.

4.6.2.4 Visibility

Visibility value within each of the cells represent the distance, from which a certain object (eg. sign, light) would be seen, with uniform smoke corresponding to the mass concentration of the smoke/soot within that cell.

\[
\text{Visibility} = \left( \frac{2.5}{D_L} \right)^{0.75},
\]

where, VFSOOT is the soot volume fraction, \( \rho_{\text{soot}} \) is the soot density, \( C_{\text{soot}} \) is the soot concentration and \( D_L \) is the optical density per meter. The parameter Visibility is calculated at cell centers and monitor points only.

4.6.2.5 Convective heat transfer coefficient: HCONV

The convective heat transfer coefficient is the proportionality constant between the heat flux and the temperature difference. HCONV is calculated at wall surfaces and monitor points.

4.6.2.6 Absorption coefficient: ABSCOF

The absorption coefficient determines how far into a medium radiation of a particular wavelength can penetrate before it is absorbed.

4.6.2.7 Radiative source term: RADSRC

Conservation of radiative energy. Within the overall energy equation, it is the divergence of the radiative heat flux that is of interest inside the medium.

4.6.2.8 Radiative heat flux: QRAD

QRAD is equal to the difference between the surface radiosity and irradiation. QRAD is calculated at wall surfaces and monitor points only. When the far-field model is enabled, it will yield results outside the DTM domain, and a combination of the DTM results and the far field results inside the DTM domain, using the DTM flux cut-off value (10 kW/m2 by default).

4.6.2.9 Far-field radiative heat flux: QRADFF

QRADFF is the far-field (three-dimensional) radiative flux without considering the blocking (shadow) effect of the geometry. The parameter QRADFF is calculated at cell centers and monitor points only. The only advantage of this parameter compared to the regular QRAD is that it allows visualising heat radiation in any grid cell in the CFD calculation domain, while the QRAD variable only shows radiation results on surfaces and in monitor points. However, as the shadow effect is ignored, the QRADFF results may be too conservative and should be used with care.
4.6.2.10 Convective heat flux: \( Q_{\text{CONV}} \)

The convective heat flux depends on the temperature difference between the surface and the adjacent fluid. It has the form

\[
Q_{\text{CONV}} = h(T_w - T_g)
\]

(4.21)

where \( Q_{\text{CONV}} \) is the convective flux normal to the surface, \( h \) is the heat transfer coefficient, \( T_w \) is the surface temperature and \( T_g \) is the gas temperature. \( Q_{\text{CONV}} \) is calculated at wall surfaces and monitor points only.

4.6.2.11 Total heat flux: \( Q \)

The total heat flux is the sum of convective and radiative heat flux. \( Q \) is calculated at wall surfaces and monitor points only.

4.6.2.12 Wall incident heat flux: \( Q_{\text{WALL}} \)

Total incident heat flux on the surface. It is the sum of the convective flux and the incident radiative flux. It does not include outgoing radiation. \( Q_{\text{WALL}} \) is calculated at wall surfaces and monitor points.

4.6.2.13 Incident radiative flux: \( Q_{\text{RINC}} \)

This is the incoming radiation from fire and/or other hot objects and surfaces. It does not include outgoing radiation. \( Q_{\text{RINC}} \) is calculated at wall surfaces and monitor points.

4.6.2.14 Heat Dose: \( Q_{\text{DOSE}} \)

Heat Dose is the time integral of the net heat flux at the selected monitor points. Heat Dose is defined by

\[
Q_{\text{DOSE}} = \int_0^t Q^{4/3} dt
\]

(4.22)

The heat dose is calculated at monitor points and surfaces.

4.6.3 3D output variables for pool simulations

The following output variables are available for simulations using FLACS-Pool.

4.6.3.1 Pool depth: \( \text{POOL}_D \)

\( \text{POOL}_D \) is the depth in meters of the pool, at different elevations (xy-plane). The \( \text{POOL}_D \) field is in addition projected to the ZLO boundary (xy-plane). The projected values for \( \text{POOL}_D \) are only for convenience, making it easier to visualize the values in the xy-plane. This is especially useful for scenarios where the pool spreads out to different elevations (z-direction).

4.6.3.2 Pool volume: \( \text{POOL}_V \)

\( \text{POOL}_V \) is the pool volume fill fraction (-), and is a derived variable, made in post-processing (Flowvis), and depends on the \( \text{POOL}_D \) variable being present in the simulation output.

4.6.3.3 Pool velocity vector: \( \text{POOL}_S \)

\( \text{POOL}_S \) is the pool velocity vector (m/s), when running the spreading pool model.
4.6 Output variables in FLACS-CFD

4.6.3.4 Pool temperature: \texttt{POOL\_T}

\texttt{POOL\_T} is the temperature in Kelvins of the pool.

4.6.3.5 Pool velocity component, x-direction: \texttt{POOL\_U}

\texttt{POOL\_U} is the pool velocity component (m/s) in x-direction, when running the spreading pool model.

4.6.3.6 Pool velocity component, y-direction: \texttt{POOL\_V}

\texttt{POOL\_V} is the pool velocity component (m/s) in y-direction, the spreading pool model.

4.6.4 General panel output variables

4.6.4.1 Panel average pressure: \texttt{PP}

This is the average pressure (Pa) acting on the panel surface in the perpendicular direction. It is the sum of the directional pressure forces acting on the panel divided by the net surface area of the panel (also accounting for the area porosity for each control volume). The sign of \texttt{PP} indicates the direction of the total force, +/- along the positive/negative direction respectively.

The panel average pressure \texttt{PP} is calculated in the following way: For each control volume face (CV face) covering the area of the panel considered, the net static pressure is calculated (static pressure on the negative side relative the coordinate axis minus the static pressure on the positive side). This net static pressure is then integrated over the blocked area of the panel and the integral is then divided by the total blocked area of the panel, to give the panel average pressure \texttt{PP}. If e.g. the area porosity of the panel is zero, the panel is totally blocked and the integral is divided by the total area of the panel. And if e.g. the panel is totally open (area porosity one, no blocked area), panel average pressure \texttt{PP} vanishes (is zero), since there is no blocked area to integrate over (the integral with respect to net static pressure over the blocked area is zero). In general the panel can be porous (partially blocked and partially open).

4.6.4.2 Panel average porosity: \texttt{PPOR}

This is the average panel porosity, it is the amount of open surface on the panel divided by the total panel area. Output of \texttt{PPOR} may be used to verify when the panel yields.

4.6.4.3 Panel average temperature: \texttt{PT}

\texttt{PT} is the panel-averaged gas temperature. It works for both active and inactive panels.

4.6.4.4 Panel average far-field radiation dose: \texttt{PQFFDOSE}

\texttt{PQFFDOSE} is the panel-averaged far-field radiation dose. It can be computed for both active and inactive panels.

4.6.4.5 Panel average far-field radiative flux: \texttt{PQRADFF}

\texttt{PQRADFF} is the panel-averaged far-field radiative flux. It is available on both active and inactive panels.

4.6.4.6 Panel average total heat flux: \texttt{PQ}

\texttt{PQ} is the panel-averaged total heat flux. It is provided for active panels and for inactive panels placed on wall surfaces. It does not work for inactive panels placed away from wall surfaces.

4.6.4.7 Panel average total incident flux: \texttt{PQTINC}

\texttt{PQTINC} is the panel-averaged total incident flux. You can obtain \texttt{PQTINC} for active panels and inactive panels placed on wall surfaces. It is not available for inactive panels placed away from wall surfaces.
4.6.5 Output variables for toxic substances

When simulating the transport of toxic gases, the following output variables give insight in the results. A description of probit functions and probability of death, is also found in Sec. 5.2 “Damage modelling” in TNO ‘Purple book’.

4.6.5.1 Toxic mole fraction TMOLE

The output variable TMOLE is the mole fraction [-] of the specified toxic species (the number of moles of the toxic species divided by the number of moles of the complete gas mixture, i.e. including air and all fuel species).

4.6.5.2 Toxic dose TDOSE

The parameter TDOSE, the toxic dose, is defined as:

\[ TDOSE(\text{mg/m}^3 \cdot \text{minute}) = \int_0^t C^n dt \]

For constant C:

\[ TDOSE(\text{mg/m}^3 \cdot \text{minute}) = C^n \cdot t \]

4.6.5.3 Toxic probit function PROBIT

The toxic probit function PROBIT is defined as:

\[ PROBIT(\text{-}) = a + b \cdot \ln(TDOSE) \]

4.6.5.4 Probability of death as function of toxic probit PDEATH

The probability of death as function of toxic probit PDEATH is defined as:

\[ PDEATH(-) = 0.5 \cdot \left[ 1 + \text{erf}\left( \frac{PROBIT - 5}{\sqrt{2}} \right) \right] \]

where:

\[ \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \]

4.6.6 Prediction of deflagration to detonation transition using DPDX and DDTLS

FLACS-CFD is able to give an indication of the likelihood of deflagration to detonation transition (DDT). It is important to know whether there is a possibility of detonation for the following reasons:

1. Detonations are more severe than deflagrations and they will propagate faster, independent of geometry. When the combustion wave propagates at a speed lower than the speed of sound, the explosion is termed deflagration. The flame, turbulence driven, typically moves with a velocity of 1000-1200 m/s, and the associated pressure wave has an amplitude of 0.01 to 10 barg. A detonation flame on the other hand, leading to shock-ignition, travels at 1500-2000 m/s, and the peak pressure can be 15-20 barg.
**Note:**

This speed (1000-1220 m/s) representing the limit of deflagrated speed, refers to the combustion products rather than the reactants; the temperature of combustion products is much higher than the room temperature and hence, as the speed of sound increases with elevating temperature, the explosion sound travels within the combustion products on a much higher speed than this of sound (in air) at room temperature; the limit of deflagrated speed itself becomes much higher than the speed of sound at room temperature (Kundu et al., 2016).

2. DDT may occur in scenarios typically seen in industrial settings, for example due to strong flame acceleration or when flame jets go through vent openings.

A functionality has been developed that uses a parameter called $\text{DPDX}$, the normalised spatial pressure gradient across the flame front, $\frac{dP}{dx}_{\text{normalised}} = \frac{dP}{dx}_{\text{actual}} \frac{X_{CV}}{P_0}$. The $\text{DPDX}$ parameter is able to indicate when the flame front captures the pressure front, which is the case when DDT occurs. Thereby the parameter gives information regarding if and where a DDT would be likely. The $\text{DPDX}$ ranges and their qualitative likelihood are shown in the figure below.

![Figure 4.9: Illustration of the $\text{DPDX}$ value range and the likelihood of DDT.](image)

The $\text{DPDX}$ value ranges indicated above are based on validation work that focused mainly on confined hydrogen explosions; they may differ for other gases and degrees of confinement, and this is a topic of on-going research and validation efforts. However, a high enough $\text{DPDX}$ value is not the only criterion that should be checked to conclude whether a detonation is likely; rather, at the same time the flame front also has to cover a large enough area compared to the detonation cell size of the gas. The reason for the size requirement is that the detonation front consists of a cell pattern, whose characteristic length scale $\lambda$ sets requirements for the minimum geometrical dimensions to enable the detonation front to propagate:

- transverse size of passage: $d > \lambda$,
- minimum distance for detonation formation: $L > 7\lambda$ (in partially obstructed pipes),
- Also possible to scale jet-initiated detonations
- Unconfined situations

To verify if the size of the region where significant values of $\text{DPDX}$ are present is “large enough” (on the order of a certain multiple of cell size), a quantitative criterion based on a length scale ratio has been developed.

![Figure 4.10: Illustration of detonation cell size and characteristic length scale.](image)
The DDT length scale ratio (DDTLS) is defined as the quotient of the characteristic geometrical dimension (LSLIM) and the detonation cell size ($\lambda$):

$$DDTLS = \frac{LSLIM}{\lambda}$$

Detonation cell size is modelled as a function of concentration, pressure and temperature. The DDTLS parameter illustrates whether the geometrical dimensions are sufficient for the propagation of a detonation front. In regions without fuel, DDTLS has the value zero. Values larger than 7 indicate that a detonation wave can propagate easily (for example in partially obstructed pipes). Typically a minimum area of 13 x 13 detonation cell sizes of (unobstructed) gas cloud ahead of the fast flame front are needed, for the DDT to be able to sustain (in an unconfined situation). For empty pipes just one cell size diameter may be required to maintain a detonation.

![Figure 4.11: Partially obstructed pipe geometry and the DPDX and DDTLS patterns indicating a likely DDT: DPDX > 5 and a sufficiently large space containing fuel ahead of the flame, as indicated by DDTLS.](image)

Cell sizes for ethylene are estimated to be 2-5 cm and for hydrogen and acetylene 1-2 cm (Shepherd et al., 1991). For natural gas, a cell size of about 30-40 cm can be assumed; in an unconfined situation, this may mean that a minimum of 4 m thickness of the stoichiometric layer will be required for a detonation to occur. See (Middha et al., 2006) and (Middha et al., 2008) for details on the DDT prediction functionality in FLACS-CFD.

### 4.6.7 Modifying names and units for output variables

It is possible to use alternative names for some output variables in FLACS-CFD. For example the DRAG name may be substituted by PDYN (dynamic pressure), and the units for the pressure variables may be set in the SINGLE FIELD VARIABLES section (Pa, hPa, kPa, barg or mbarg).

To change the unit from barg to Pa for pressure variable $P$ (pressure) open the cs-file in a text editor, locate the SINGLE FIELD VARIABLES section and modify as follows:

Output in barg (default):
NP "P " 1 "(barg) " N  
"Pressure"

Output in Pa:
NP "P " 1 "(Pa) " N  
"Pressure"

The following table lists the variables where the units can be changed by using the method above.

Table 4.6: Modified variables names.

<table>
<thead>
<tr>
<th>ID</th>
<th>Default name</th>
<th>Alternative name</th>
<th>Default units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>P</td>
<td>-</td>
<td>(barg)</td>
<td>Static pressure</td>
</tr>
<tr>
<td>NP_IMP</td>
<td>P_IMP</td>
<td>-</td>
<td>(Pa+s)</td>
<td>Static pressure impulse</td>
</tr>
<tr>
<td>NUDRAG</td>
<td>UDRAG</td>
<td>UPDYN</td>
<td>(Pa)</td>
<td>Dynamic pressure x-component</td>
</tr>
<tr>
<td>NVDRAG</td>
<td>VDRAG</td>
<td>VPDYN</td>
<td>(Pa)</td>
<td>Dynamic pressure y-component</td>
</tr>
<tr>
<td>NWDRAG</td>
<td>WDRAG</td>
<td>WPDYN</td>
<td>(Pa)</td>
<td>Dynamic pressure z-component</td>
</tr>
<tr>
<td>NDRAG</td>
<td>DRAG</td>
<td>PDYN</td>
<td>(Pa)</td>
<td>Dynamic pressure</td>
</tr>
<tr>
<td>NUDRAG_IMP</td>
<td>UDRAG_IMP</td>
<td>-</td>
<td>(Pa+s)</td>
<td>Dynamic pressure impulse x-component</td>
</tr>
<tr>
<td>NVDRAG_IMP</td>
<td>VDRAG_IMP</td>
<td>-</td>
<td>(Pa+s)</td>
<td>Dynamic pressure impulse y-component</td>
</tr>
<tr>
<td>NWDRAG_IMP</td>
<td>WDRAG_IMP</td>
<td>-</td>
<td>(Pa+s)</td>
<td>Dynamic pressure impulse z-component</td>
</tr>
<tr>
<td>NDRAG_IMP</td>
<td>DRAG_IMP</td>
<td>-</td>
<td>(Pa+s)</td>
<td>Dynamic pressure impulse</td>
</tr>
<tr>
<td>NPTOT</td>
<td>*</td>
<td>PTOT</td>
<td>(Pa)</td>
<td>Total pressure</td>
</tr>
<tr>
<td>NPTOT_IMP</td>
<td>*</td>
<td>PTOT_IMP</td>
<td>(Pa+s)</td>
<td>Total pressure impulse</td>
</tr>
</tbody>
</table>

Remarks:

* = Not available in default template, but can be added manually.

Note:

The ID (the first word in the variable specification) cannot be changed. The variable names can be maximally 32 and the unit text maximally 16 characters long.

4.7 Files in FLACS-CFD

Input and output data for FLACS-CFD are stored in files. The name of each file consists of two parts separated by a dot (.). The first part of the file name contains a two-letter prefix followed by the 6-digit job number. The second part of the file name, called the file type, contains a prefix of one or more letters followed by one or more digits (dat3 for most of the files, n001 etc. for the leak data files).

Summary of the files used by FLACS-CFD:

Table 4.7: Summary of files in FLACS-CFD.

<table>
<thead>
<tr>
<th>File name</th>
<th>Contents of file</th>
</tr>
</thead>
<tbody>
<tr>
<td>cs000000.dat3</td>
<td>Scenario</td>
</tr>
<tr>
<td>File name</td>
<td>Contents of file</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>cg000000.dat3</td>
<td>Grid</td>
</tr>
<tr>
<td>cp000000.dat3</td>
<td>Porosities</td>
</tr>
<tr>
<td>co000000.dat3</td>
<td>Obstructions</td>
</tr>
<tr>
<td>cc000000.dat3</td>
<td>Runtime simulation control</td>
</tr>
<tr>
<td>cn000000.dat3</td>
<td>Time dependent CFL-numbers</td>
</tr>
<tr>
<td>cl000000.n000</td>
<td>Time dependent leak data</td>
</tr>
<tr>
<td>cl000000.POOL</td>
<td>Time dependent pool leak data</td>
</tr>
<tr>
<td>r1000000.dat3</td>
<td>Scalar-time output</td>
</tr>
<tr>
<td>r3000000.dat3</td>
<td>Field output</td>
</tr>
<tr>
<td>rt000000.dat3</td>
<td>Simulation log</td>
</tr>
<tr>
<td>tt000000</td>
<td>Simulation log, terminal printout</td>
</tr>
<tr>
<td>rd000000.n000</td>
<td>Simulation dump</td>
</tr>
<tr>
<td>rx000000.n000</td>
<td>Simulation save, temporary file created by FLACS-CFD</td>
</tr>
</tbody>
</table>

Summary of the identification numbers used in the files:

Table 4.8: Identification numbers

<table>
<thead>
<tr>
<th>File name</th>
<th>Meaning of file type digits</th>
</tr>
</thead>
<tbody>
<tr>
<td>??000000.dat3</td>
<td>Number of dimensions (1, 2, or 3)</td>
</tr>
<tr>
<td>cl000000.n000</td>
<td>Number identifying a leak</td>
</tr>
<tr>
<td>rd000000.n000</td>
<td>Number identifying a Dump or Load</td>
</tr>
<tr>
<td>rx000000.n000</td>
<td>Number identifying a variable</td>
</tr>
</tbody>
</table>

4.7.1 Input files for FLACS-CFD simulations

This section summarises the various input files for FLACS-CFD.

4.7.1.1 Geometry files

Defines the geometry, generated by CASD, required by porosity calculators and Flowvis. Further details can be found in section Geometry in FLACS-CFD. For briefness this file may be called the co-file or obstruction file hereafter.

File name template: co000000.dat3 or co000000.geo

4.7.1.2 Grid file

Defines the computational mesh, generated by CASD, required by FGC.

File name template: cg000000.dat3

For briefness this file may be called the cg-file or grid file hereafter. It is a binary file and will be created when using the grid definition menu in CASD. The grid file stores the computational grid, i.e. the discrete representation of the simulation volume. The simplest form of a grid is a uniform grid, where all control volumes have the same size and shape.

The figure below shows a two-dimensional section of a uniform grid, and illustrates the position of the internal nodes and the boundary nodes relative to the grid lines.
4.7 Files in FLACS-CFD

The numbering scheme for the nodes and grid lines may need some attention when working with the data directly. The table below shows some details of the numbering scheme:

<table>
<thead>
<tr>
<th>Sizes</th>
<th>Internal nodes</th>
<th>Boundary nodes</th>
<th>Grid lines</th>
<th>Used in</th>
</tr>
</thead>
<tbody>
<tr>
<td>N = NX, NY, NZ</td>
<td>2 to N-1</td>
<td>1 and N</td>
<td>2 to N</td>
<td>Flacs and CASD</td>
</tr>
</tbody>
</table>

In typical semi-confined geometries like an offshore module one may create a grid only covering the interior including the outer walls of the module, but it is recommended to extend the grid to improve the boundary conditions. In the cases of very open geometries one should always extend the grid well outside the main explosion area.

More information can be found in sections FGC (Flacs Geometry Calculator) and Calculate porosities.

4.7.1.3 Header file

Defines the cs, co, cg and cm files to be used by CASD.

File name template: 000000.caj

This file is created by CASD and is a text file listing the cs, co, cg and cm files in the current simulation job file set. The file is not used by any other FLACS-CFD programs or utilities, and is not necessary to run simulations.

4.7.1.4 Porosity file

Defines the porosities for each grid cell, calculated by the program FGC (Flacs Geometry Calculator) from the co and cg-files, required by Flacs.
File name template: cp000000.dat3
For briefness this file may be called the cp-file or porosity file hereafter. It is a binary file and is created by
FGC (Flacs Geometry Calculator).
It contains data which are calculated based on the geometry and the grid. FLACS-CFD will stop if the cp-file
is not accessible for reading. The cp-file is quite large, the size (in bytes) may be calculated as follows:

\[ SIZE = 10 \cdot 4 \cdot NX \cdot NY \cdot NZ \]

4.7.1.5 Scenario file

Defines the general scenario (monitor points, output variables, fuel region, vents, ignition position, etc.),
required by Flacs.
File name template: cs000000.dat3
For briefness this file may be called the cs-file or scenario file hereafter. It is a text file and will be created
when using the scenario definition menu in CASD.
The term scenario was defined in the introduction to this User Manual as the set of parameters which may
be used to control the behaviour of a given FLACS-CFD simulation.
The first line of the scenario file identifies the file format, for FLACS-CFD 22.2 this is set to the following
text string: "VERSION 0.5". This must not be changed manually since it will be used by FLACS-CFD to
determine how to read and interpret the file.
Only the most recent format is described here.
The scenario file contains several sections, which are structured as follows:

SECTION_NAME
  A section may contain several lines ...
EXIT SECTION_NAME

The sections in the scenario file are:

SINGLE_FIELD_VARIABLES
MONITOR_POINTS
PRESSURE_RELIEF_PANELS
SINGLE_FIELD_SCALAR_TIME_OUTPUT
SINGLE_FIELD_3D_OUTPUT
SIMULATION_AND_OUTPUT_CONTROL
BOUNDARY_CONDITIONS
INITIAL_CONDITIONS
GAS_COMPOSITION_AND_VOLUME
VOLUME_FRACTIONS
IGNITION
POOL
LEAKS
OUTLET / VESSEL
WATERSPRAYS
LOUVRE_PANELS
GRATING
MONITOR_VOLUMES

4.7.1.6 Setup file

The setup file is an optional file used to access certain advanced settings that govern, for example, the choice
of specific models and solver variants. The setup file contains pure text and can be prepared in any text
editor or, more easily, created by CASD when the setup section in the scenario menu is filled in.
The FLACS-CFD RunManager detects the setup file automatically if the file exists in the same directory as
the regular job files and the file name follows the pattern cs<jobno>.SETUP. If the file is named like this, it
will also be read automatically when using the run runflacs command. If the setup file has a different
name then it must be supplied as argument #2, i.e. after the job number:
4.7 Files in FLACS-CFD

> run runflacs 010101 cs010100.SETUP

or with the detailed syntax

> run runflacs 010101 setup=cs010100.SETUP

which allows adding options for the parallel solver as well. The setup file may contain the following so-called namelists: JOBSPEC, SETUP and PARAMETERS. The namelists may be entered in any order or may alternatively be left out entirely.

```
VERSION 1.1
$JOBSPEC
  ...
$END
$SETUP
  ...
$END
$PARAMETERS
  ...
$END
```

The $ in the name list must be positioned in column 2 (only on certain machine types).

### 4.7.1.6.1 The JOBSPEC namelist

The available keywords in JOBSPEC and their default values are summarised below:

```
VERSION 1.1
$JOBSPEC
  KEEP_OUTPUT = .FALSE.
  RESET_LOAD = .TRUE.
$END
```

The meaning of the keywords is as follows:

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEEP_OUTPUT</td>
<td>Keep old results in the existing r1-file and r3-file (append new results):</td>
</tr>
<tr>
<td></td>
<td>• .TRUE. if you want to run a continuation run</td>
</tr>
<tr>
<td></td>
<td>• .FALSE. if you restart a new scenario</td>
</tr>
<tr>
<td>RESET_LOAD</td>
<td>When set to .TRUE., resets at LOAD time the maximum (reference) fuel amount;</td>
</tr>
<tr>
<td></td>
<td>this can be used together with the NPLOT parameter to determine how often</td>
</tr>
<tr>
<td></td>
<td>data for field plots is written to file during a simulation; this can be</td>
</tr>
<tr>
<td></td>
<td>useful for starting an explosion after a dispersion simulation.</td>
</tr>
</tbody>
</table>

### 4.7.1.6.2 The SETUP namelist

The available keywords in SETUP and their default values are summarised below:

```
VERSION 1.1
$SETUP
  COMBUSTION_MODEL = "BETA3"
  TURBULENCE_MODEL = "K-EPS"
  GASDATA_MODEL = "DEFAULT"
$END
```

These controls will assume default values if not specified by the user. The meaning of the keywords is as follows:
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COMBUSTION_MODEL</td>
<td>Selection of combustion model:</td>
</tr>
<tr>
<td></td>
<td>• BETA3</td>
</tr>
<tr>
<td></td>
<td>• EDC (FLACS-Fire)</td>
</tr>
<tr>
<td>TURBULENCE_MODEL</td>
<td>Selection of turbulence model:</td>
</tr>
<tr>
<td></td>
<td>• K-EPS</td>
</tr>
<tr>
<td>GASDATA_MODEL</td>
<td>Selection of gasdata model, laminar burning velocity as function of ER for the fuels:</td>
</tr>
<tr>
<td></td>
<td>• DEFAULT</td>
</tr>
<tr>
<td></td>
<td>• name of directory containing gasdata files, cf. Example using a modified gasdata file for dispersion</td>
</tr>
<tr>
<td></td>
<td>• BUILT-IN</td>
</tr>
<tr>
<td>AIR</td>
<td>Specification of O2 fraction in air ((&lt;\text{percent}&gt;) is the fraction multiplied by 100):</td>
</tr>
<tr>
<td></td>
<td>• NORMAL</td>
</tr>
<tr>
<td></td>
<td>• (&lt;\text{percent}&gt;%\text{VOLUME})</td>
</tr>
<tr>
<td></td>
<td>• (&lt;\text{percent}&gt;%\text{MOLE})</td>
</tr>
<tr>
<td></td>
<td>• (&lt;\text{percent}&gt;%\text{MASS})</td>
</tr>
<tr>
<td></td>
<td>Specification of relative humidity in air (only available in pool simulation type of FLACS-CFD):</td>
</tr>
<tr>
<td></td>
<td>• HUMIDITY = (&lt;\text{RH}&gt;%\text{,@}&lt;\text{TEMPERATURE}&gt;{\text{C,K}})</td>
</tr>
<tr>
<td></td>
<td>• Relative humidity, (&lt;\text{RH}&gt;) can be specified in percent (0-100) or fraction (0-1), at specified (&lt;\text{TEMPERATURE}&gt;) in Celsius or Kelvin:</td>
</tr>
<tr>
<td></td>
<td>– AIR = &quot;NORMAL,HUMIDITY=90%20\text{C}&quot; (90 percent at 20 degrees Celsius)</td>
</tr>
<tr>
<td></td>
<td>– AIR = &quot;NORMAL,HUMIDITY=0.9@293\text{K}&quot; (0.9 fraction water at 293 Kelvin)</td>
</tr>
<tr>
<td></td>
<td>• Default temperature is 0 degrees Celsius.</td>
</tr>
<tr>
<td></td>
<td>• The output variable FOG gives the mass fraction of water in the liquid phase.</td>
</tr>
<tr>
<td></td>
<td>Refer to section Humidity and fog for technical details. Note that AIR is included in the cs-file and the string provided there will overwrite the one from the setup file! Therefore AIR in the setup file is deprecated.</td>
</tr>
</tbody>
</table>
### 4.7 Files in FLACS-CFD

#### 4.7.1.6.3 Time stepping for ventilation simulations

It is possible to choose a time stepping algorithm which only includes the convective speed, by specifying `STEP "V_MIN=<real number>"`. The CFL-number based on the convective speed (`CFLV`) is given as usual (in the cs-file). The CFL-number based on the speed of sound (`CFLC`) is not used (i.e. the value of `CFLC` given in the cs-file is not employed). Acoustical waves are not resolved using this approach.

This criterion is intended as an alternative to the default criterion when the flow is stationary/slowly varying or nearly incompressible. It has been tested in wind/ventilation simulations. It should not be used for an explosion simulation. A speed-up factor of ca. 8 is seen from test simulations of wind/ventilation using this criterion compared to the default setup in FLACS-CFD. However, the speed-up factor depends on the scenario.

When high-momentum leaks are modelled, the convective speed is relatively large, and the speed-up effect when using this criterion may be limited in this case. To employ this criterion, specify a velocity `V_MIN [m/s]`, as argument to `STEP` (see Timestep code) for example, `STEP "V_MIN=1.0"`. This velocity, `V_MIN`, is used by the time stepping algorithm as a minimum speed when determining the time step, its value should be positive and not too small.

If a wind field is specified, a natural choice would be to set `V_MIN` equal the value of `WIND_SPEED`, if no wind field is specified a value of 1.0 m/s for `V_MIN` would be a typical value. The value of `V_MIN` is in general only used in the initial phase of the simulation (when normally a flow field is started from a condition at rest to ensure that the time step is not too large even though the velocity of the flow is zero or very small (when the default time step criterion is used the value of `CFLC` limits the time step even when the convective speed is zero).

**Attention:**

The intention of using this time step criterion is to speed up the calculation by using a coarser resolution in time (longer time steps), and this may change the simulation results compared to a finer resolution in time (smaller time steps) when the flow field is transient. A typical choice of `CFLV` would be 1.0, a larger value of `CFLV` may lead to unstable results (depending on the scenario).

#### 4.7.1.6.4 Example using a modified gasdata file for dispersion

In the following example, pure hexane will be used as fuel in FLACS-CFD to simulate dispersion of a fuel species with molecular weight close to the molecular weight of hexane, when released in air. In this example it is assumed that the mentioned fuel species has a flammable range from 3% molar fraction of fuel in the fuel-air mixture (in this example regarded as the Lower Flammability Limit) to 12% molar fraction (in this example regarded as the Upper Flammability Limit). Further, it is assumed that the scenario involves dispersion only, i.e. no combustion is involved, and that the variable of interest is the integrated volume of the fuel-air concentration within the flammable range (over the whole or part of the computational domain) rather than the mass of fuel with fuel-air concentration within the flammable range.

Steps to set up such a scenario:

1. Calculate the equivalence ratio `ER`, corresponding to the molar fraction 3% of fuel hexane in fuel-air mixture, and denote as the value `ER_LFL`. The equivalence ratio `ER` is defined as the ratio of the mass

---

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMBIENT_PRESSURE</td>
<td>Specification of ambient pressure, the default value is:</td>
</tr>
<tr>
<td></td>
<td>• 1.0E5</td>
</tr>
<tr>
<td></td>
<td>Note that AMBIENT_PRESSURE is included in the cs-file and the value provided there will overwrite the one from the setup file! Therefore AMBIENT_PRESSURE in the setup file is deprecated.</td>
</tr>
<tr>
<td>PS1</td>
<td>Removes the criterion in FLACS-CFD that the maximum speed cannot exceed sonic speed:</td>
</tr>
<tr>
<td></td>
<td>• 01</td>
</tr>
</tbody>
</table>
of fuel to the mass of oxygen (for the fuel-air mixture considered), divided by the ratio of mass of fuel to mass of oxygen at the stoichiometric concentration.

2. Calculate the equivalence ratio $ER$, corresponding to the molar fraction 12% of fuel hexane in fuel-air mixture, and denote this as the value $ER_{UFL}$.

3. Go to the working directory with your FLACS-CFD dispersion simulation, and copy the gasdata file $CO2$ from the standard gasdata library to the current working directory, naming the copied file $HEXANE$ (here commands for Linux operating system are shown, it can be done in a similar way on computers with a Windows operating system):

```bash
> cp /usr/local/Gexcon/FLACS-CFD_22.2/bin/files/gasdata20.1/CO2 HEXANE
```

Here `/usr/local/` is the default installation path for FLACS-CFD on Linux (this may be different depending on your installation).

4. Modify (using a text editor) the new text file named $HEXANE$ so that its content becomes:

```plaintext
# ER SLAM SIM
#--- --- --------- ---------
#
<$ER_{LFL}$> 0.0 0.0
<$\frac{(ER_{LFL}+ER_{UFL})}{2.0}$> 0.0 0.0
<$ER_{UFL}$> 0.0 0.0
```

Here $<$ER$_{LFL}$> is replaced by the calculated value of $ER_{LFL}$, $<$ER$_{UFL}$> is replaced by the calculated value of $ER_{UFL}$, and $<$\frac{(ER$_{LFL}$+ER$_{UFL}$)}{2.0}$> is replaced by the mean value of $ER_{LFL}$ and $ER_{UFL}$.

5. Define the gas monitor region and other input parameters for your simulation, including a setup file (in this example the setup file is named cs123456.SETUP) with the content:

```plaintext
VERSION 1.1
$SETUP
   GASDATA_MODEL = "./"
$END
```

Here "./" means that FLACS-CFD should start looking for gasdata file in the current directory (this is assumed to be your working directory where the FLACS-CFD simulation files for your job no. are stored).

6. Start the Flacs simulator using your setup file

```bash
> run runflacs 123456 cs123456.SETUP
```

In this example the job number is 123456, and run is an alias for:

```bash
> /usr/local/Gexcon/FLACS-CFD_22.2/bin/run
```

7. You can check from the text log file (in this example rt123456.dat3 or tt123456) that the effective flammability limits expressed in molar fractions (mole of fuel divided by mole of fuel-air mixture) have the intended values. For example, you should find the following information regarding the Lower Flammability Limit reported in the text log file:

```
Flammability limits, mole fractions:
LFL \%LFL 0.030 100.0
```

and for the Upper Flammability Limit:

```
UFL \%LFL 0.120 400.0
```
4.7.1.6.5 Modifying laminar burning velocities in the gasdata file for explosion simulations

For explosion simulations, it can sometimes be necessary to define laminar burning velocities in the gasdata-file read by FLACS-CFD, for example when considering initial pressures significantly below atmospheric values or inerted atmospheres. Another possible application can be to check the effect of using a different set of measurements for the laminar burning velocity than the FLACS-CFD default. As an example, the file PROPANE may be copied (as in the above example) to a new directory, which is then set as GASDATA_MODEL directory. Implementing the laminar burning velocities for propane given by Law (1980), the PROPANE file could contain:

```
# ER SLAM SIM
#-- --- ---------- ----------
#  0.52  0.00e-2  6.471e-2
0.60  12.50e-2  13.826e-2
0.70  23.40e-2  22.219e-2
0.80  31.20e-2  30.863e-2
0.90  37.60e-2  38.890e-2
1.00  42.40e-2  45.213e-2
1.10  43.80e-2  48.612e-2
1.20  41.00e-2  48.145e-2
1.30  31.90e-2  42.428e-2
1.40  23.30e-2  30.025e-2
1.50  14.40e-2  16.281e-2
1.60  10.00e-2  10.900e-2
1.70  7.72e-2   7.723e-2
1.80  5.58e-2   5.577e-2
1.90  4.06e-2   4.057e-2
2.00  2.96e-2   2.959e-2
2.10  1.80e-2   2.116e-2
2.20  1.00e-2   1.521e-2
2.30  0.50e-2   1.101e-2
2.40  0.00e-2   0.801e-2
```

Attention:

To determine SLAM at intermediate ER values, Flacs uses an interpolation routine. The interpolation can lead to unexpected oscillations in the resulting laminar burning velocities if the given ER/SLAM points do not result in a sufficiently smooth curve. As a basic test, the laminar burning velocity for the initial equivalence ratio should be checked in the rt/tt-file.

In general, replacing the FLACS-CFD default values of the gas properties means that the validation lapses and should therefore only be done when examining effects like those mentioned above and when reliable experimental data is available.

Note:

The SIM parameters in the gasdata file are no longer used by the models in the Flacs simulator; they can be set to 0.0 if a gasdata file is prepared manually.

4.7.1.6.6 The PARAMETERS namelist

The available keywords in PARAMETERS and their default values are shown below:

```
VERSION 1.1
$PARAMETERS
  ZERO_APOR = 0.0
  ZERO_VPOR = 0.0
  ER_LOW = 0.0
  ER_HIGH = 0.0
  FLUX_CONTROL = 1
$END
```

The values in the PARAMETERS namelist are numerical, not text strings as in the SETUP namelist. The meaning of the keywords in the PARAMETERS namelist is as follows:
<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZERO_APOR</td>
<td>Lower limit for area porosities, may be used to avoid problems associated with small area porosities:</td>
</tr>
<tr>
<td></td>
<td>• 0.0 not in effect (default)</td>
</tr>
<tr>
<td></td>
<td>• 0.05-0.1 to avoid MASS_RESIDUAL</td>
</tr>
<tr>
<td>ZERO_VPOR</td>
<td>Lower limit for volume porosities, may be used to avoid problems associated with small volume porosities:</td>
</tr>
<tr>
<td></td>
<td>• 0.0 not in effect (default)</td>
</tr>
<tr>
<td></td>
<td>• 0.05-0.1 to avoid MASS_RESIDUAL</td>
</tr>
<tr>
<td>ER_LOW</td>
<td>Lower bound of the ER range reported for the GAS_MONITOR_REGION defined in a cs-file.</td>
</tr>
<tr>
<td></td>
<td>• 0.0 not in effect (using ER_LFL as default)</td>
</tr>
<tr>
<td></td>
<td>See the section about the gas monitor region for more details.</td>
</tr>
<tr>
<td>ER_HIGH</td>
<td>Upper bound of the ER range reported for the GAS_MONITOR_REGION defined in a cs-file.</td>
</tr>
<tr>
<td></td>
<td>• 0.0 not in effect (using ER_UFL as default)</td>
</tr>
<tr>
<td></td>
<td>See the section about the gas monitor region for more details.</td>
</tr>
<tr>
<td>FLUX_CONTROL</td>
<td>Controlling oscillating velocities on stretched grids:</td>
</tr>
<tr>
<td></td>
<td>• 1 = normal (default)</td>
</tr>
<tr>
<td></td>
<td>• 2 = reduced to avoid oscillations</td>
</tr>
</tbody>
</table>

Changing the values in the PARAMETERS namelist will affect the accuracy and stability of the code. In cases where the simulation gives MASS_RESIDUAL problems it may be beneficial to set the values of ZERO_APOR=0.1 and ZERO_VPOR=0.1 (or similar values in the order of 0.01 - 0.1).

In cases with stretched grids one may see oscillating flow in where the ratio between smallest and largest side length of the control volume is large, try to set FLUX_CONTROL=2 to avoid the problem.

There may be additional keywords in the PARAMETERS namelist, which are intended for development projects and should not be employed in production use.

### 4.7.1.6.7 Examples of using a setup file to define a high-pressure-temperature region

In Flacs it is possible to define a region with given pressure and/or temperature; in addition the region can contain flammables (used for special shape clouds). To define a high-pressure-temperature region in Flacs a setup file in the general form as shown below must be provided (examples follow):

```plaintext
VERSION 1.1
$PARAMETERS

PFAC = 

HPPOS = 

HPSIZ = 

HPEXP = 

HPTYP = 

$END

$SETUP

    KEYS="PS1=",P_SET=Q,"T_SET=Q",HPCON=""

$END
```
The variables are defined as follows:

- **PFAC** is used to change the pressure and density in the defined region, if not **P_SET** and **T_SET** are applied (see also table *Switches in P_SET and T_SET*). For example, **PFAC = 2** will multiply the pressure and density by 2.

- **HPPOS** is the lower x, y, z coordinates of the region, not the center coordinates.

- **HPSIZ** is the size of the region in the x, y, z direction.

- **HPEXP** is the exponents of a general ellipsoid, default values are 2.0, 2.0, 2.0: 
  \[(x/a)^2 + (y/b)^2 + (z/c)^2 = 1.\]

- **HTYP** is related to the coefficients \((a, b, c)\) of a general ellipsoid, for example: 1, 1, 1 = sphere, 1, 1, 0 = z-direction cylinder, 0, 1, 1 = x-direction cylinder.

- **PS1** is related to supersonic flow and should be set to 01 (default is 11) if one expects such flow velocities. More details on **PS1** are given in the *Special control keys* section.

- **P_SET** is the pressure in the region (bara).

- **T_SET** is the temperature in the region (degrees Celsius).

- **Q(y or -)** are switches in **P_SET** and **T_SET**, explained in the table *Switches in P_SET and T_SET*, with examples.

- **HPCON** is the confinement/fuel in the region, and is defined by XYZF which have the following values (one for each direction XYZ):
  - **XYZ** possible values:
    - `'-'` (negative half),
    - `'+'` (positive half) or
    - `='` (both halves)
  - **F** can have the following values (FUEL lean or FUEL rich):
    - `'0'` (lean, i.e. ER9) or
    - `'1'` (rich, i.e. ER0) concentration of fuel

<table>
<thead>
<tr>
<th>P_SET</th>
<th>T_SET</th>
<th>P</th>
<th>T</th>
<th>RHO (density)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P_SET=-:11</td>
<td>T_SET=-:25</td>
<td>P = P*PFAC</td>
<td>T = unchanged</td>
<td>RHO = RHO*PFAC</td>
</tr>
<tr>
<td>P_SET=Y:11</td>
<td>T_SET=-:25</td>
<td>P = 11 bara</td>
<td>T = P/(R*RHO)</td>
<td>RHO = unchanged</td>
</tr>
<tr>
<td>P_SET=-:11</td>
<td>T_SET=Y:25</td>
<td>P = unchanged</td>
<td>T = 25 °C</td>
<td>RHO = P/(R*T)</td>
</tr>
</tbody>
</table>

**Warning:**

It is recommended to use small time steps (e.g. **CFLC = CFLV = 0.1 - 0.025**) if the region has a high initial pressure and/or temperature compared to ambient conditions. The grid should be extended far away from the “bursting” vessel since numerical artifacts may be seen when the blast wave reaches the boundary. A relatively fine grid resolution is also recommended. Grid sensitivities should be performed.

**Example with a bursting vessel**
To simulate a bursting vessel (with air) use for instance a cylinder-shaped high-pressure-temperature region. Here, the diameter of the cylinder is smaller than the length of the cylinder. For such cases you should have at least 6-10 cells across the diameter of the high-pressure-temperature region (the smallest geometric length), and the same grid spacing along the length of the cylinder (in other directions). Perform a few simulations to check the grid sensitivity for your FLACS-CFD study: when you have defined the base grid according to the above guideline, test the sensitivity with respect to grid resolution for the important simulation parameters by employing a grid with, e.g., twice as fine resolution as the base grid.

The following setup file will define a 12m diameter spherical high-pressure-temperature region, at 300 barg and 2500 °C, respectively.

```
VERSION 1.1
$PARAMETERS
   PFAC = 1
   HPPOS = 44, 44, 0
   HPSIZ = 12, 12, 12
   HPEXP = 2.0, 2.0, 2.0
   HPTYP = 1, 1, 1
$END

$SETUP
   KEYS="PS1=01,P_SET=Y:301,T_SET=Y:2500"
$END
```

This setup file may be used with the standard Flacs solver, or the FLACS-Blast solver. The Blast solver in Flacs (see more details about the solver in sections Applications and Modelling and application limitations) is 4-5 times faster than the Flacs solver, requires much less memory (i.e. larger jobs can be simulated), but does not have panel and porosity functionality, and no combustion capability. To use the Blast solver for this scenario: in CASD, choose a BLAST template, define a setup file as shown and remove EXPLOSIVE from the scenario file (otherwise this will overwrite the KEYS settings).

**Example with a pressurised vessel with evaporating liquid**

Rupture of a pressurised vessel, filled with evaporating liquid (which may become a boiling liquid expanding vapor explosion, BLEVE), may be set up in the following way:

1. Transfer all liquid into pressurised gas.
2. Use a vessel of dimensions corresponding to the correct pressure and boiling point temperature.

If, for instance, only a z≥0 hemisphere of fuel is wanted, one can change the KEYS string accordingly:

```
KEYS="PS1=01,P_SET=Y:301,T_SET=Y:2500,HPCON==+1"
```

**Example with a hemispherical cloud**

A hemispherical cloud (in this example hemispherical in the sense that points of the hemisphere always have z-coordinate larger than or equal to zero) at ambient T and P with diameter 20m, and center (for the xy-plane) in x=0, y=0, can be defined like this:

```
VERSION 1.1
$PARAMETERS
   PFAC = 1
   HPPOS = -10, -10, -10
   HPSIZ = 20, 20, 20
   HPEXP = 2.0, 2.0, 2.0
   HPTYP = 1, 1, 1
$END
```
4.7 Files in FLACS-CFD

4.7.1.7 Thermal Active Walls

The optional TWALL file can be used to define thermally active walls.

File name template: cs000000.TWALL

Some walls can be thermally active, which means that they conduct heat and change temperature over time. A wall is declared as thermally active by adding it and its material in a cs000000.TWALL file.

The input format is as follows:

- **MAT**: tells that the next lines contain information about the wall materials
  
  - Predefined materials are STEEL and CONCRETE. (See Table material properties)
  
  - Up to ten user-specified materials are accepted. For user-specified materials information about density (DENS, kg/m$^3$), specific heat (CP, J/(kgK)), conductivity (COND, W/(mK)) and initial temperature (T, C) is needed.
  
  Example: {USER1, CP=840.0, DENS=1600, COND=0.69, T=10}

- **BOXES**: tells that thermal walls are going to be specified.

- Walls must be given as follows, Material: $x_0 y_0 z_0 x_l y_l z_l$, where subscript 0 denotes a corner point and subscript l the dimension.
  
  Example: {USER1: 0.0 0.0 0.0 12.75 0.1 5.9}

An example of an input file for thermal active walls is as follows:

MAT
STEEL
USER1, CP=840.0, DENS=1600, COND=0.69, T=10
BOXES
USER1: 0.0 0.0 0.0 12.75 0.1 5.9
USER1: 0.0 5.5 0.0 12.75 0.1 5.9
STEEL: 0.0 0.0 5.9 12.75 5.5 0.1
USER1: 12.65 0.0 0.0 0.1 5.5 5.9
USER1: 0.0 0.0 2.81 0.1 5.5 3.09
USER1: 0.0 4.49 0.0 0.1 1.01 2.81

4.7.1.8 Cloud file

Optional file used to define fuel clouds of arbitrary shape.

File name template: cs000000.CLOUD

The cloud interface module in FLACS-CFD can be used to specify rectangular or other than rectangular shapes of clouds with uniform or non-uniform concentration of fuel. The cloud interface module is automatically invoked if FLACS-CFD finds the file cs000000.CLOUD in the working directory, where 000000 is a given 6-digit job number.

When a simulation is started from scratch (not from a dump file), the fuel cloud detected from the cs000000.CLOUD file comes in addition to the ordinary fuel region defined in the cs000000.dat3 file. If you intend to have no contribution from the ordinary fuel region (since you want the fuel cloud to be specified only from input given in the cs000000.CLOUD file), this can be ensured by setting

```
POSITION_OF_FUEL_REGION 0.0, 0.0, 0.0
DIMENSION_OF_FUEL_REGION 0.0, 0.0, 0.0
```
in the ordinary scenario file cs000000.dat3.
When a simulation is started from a dump file (e.g. rd000000.n001), information about the fuel-air cloud is not taken from the cs000000.dat3 file, but by default from the information about the fuel-air cloud included in the dump file. However, if a cloud file, cs000000.CLOUD, is present, then the information about the fuel-air cloud included in the cloud file, has precedence over the information included in the dump file. Here the enthalpy field is consistently converted (this is automatically done) taking into account the change of fuel concentration in the cs000000.CLOUD file compared to the dump file (works in a similar way as for the option "convert" that can be used when the utility program rdfile is invoked).

A simple example of a CLOUD file defining a 1.5 m x 1.5 m x 0.34 m rectangular cloud of homogeneous concentration, is shown below:

```
VERSION 1.0
CV_SIZE 1.0
CV_RESOLUTION 3

POINT:
  9.2500 -0.8500 0.0240 1.0000
  10.7500 -0.8500 0.0240 1.0000
  10.7500 0.6500 0.0240 1.0000
  9.2500 0.6500 0.0240 1.0000
  9.2500 -0.8500 0.3640 1.0000
  10.7500 -0.8500 0.3640 1.0000
  10.7500 0.6500 0.3640 1.0000
  9.2500 0.6500 0.3640 1.0000
```

In this file, the first line:

```
VERSION 1.0
```

is a version identification for future compatibility checking and the value of 1.0 should be entered to be consistent with future interpretation. The next (non-empty) line defines the type of data object used:

```
POINT
```

The following data objects can be used with the cloud interface:

- POINT: the next lines contain points :N*(x,y,z,f)
- TETRAHEDRON: the next lines contain points :4*(x,y,z,f)
- HEXAHEDRON: the next lines contain points :8*(x,y,z,f)
- ARRAY: the next lines contain points :NI*NJ*NK*(x,y,z,f)
- FLUENT-UNS: the next lines contain FLUENT-UNS profile data
- (: the start of FLUENT-UNS profile data

The sequences of coordinates follow the type of data object. The coordinate of the first point is:

```
9.2500 -0.8500 0.0240 1.0000
```

where the definition of each of the four numbers is:

- x = x-coordinate
- y = y-coordinate
- z = z-coordinate
- f = fuel mass fraction
4.7 Files in FLACS-CFD

Note:

The effective fuel concentration modelled in the core simulator flacs, will always have an equivalence ratio (ER) within the range between ER9 and ER0, even when the value of f corresponds to an equivalence ratio outside this range. Let us consider the following example where the fuel composition consists of pure methane, ER9 equals 0.0 (corresponding to pure air) and ER0 equals 1.0 (corresponding to stoichiometric mixture of fuel and air). And the mass fraction of methane in a stoichiometric methane-air mixture is 0.055047 (5.5047%). In this example the reported maximum value of FUEL (mass fraction of fuel in fuel-air mixture) is 0.055047 (stoichiometric condition) even when a fuel region is defined in a cs000000.CLOUD file specifying f equal 1.0 (corresponding to pure fuel).

Several point sets, or (more generally) several data object sets, can be specified by initiating the sequence of coordinates by a :

as on the fourth line of the example.

Additional options are available with the cloud interface, such as filters:

- ER_FLAT a b

This option filters the f-field with a cutoff value (a) and an insert value (b). If the option ‘ER_FLAT’ is not specified the program does:

```plaintext
if (ERf < ER9) ERf = ER9; else if (ERf > ER0) ERf = ER0
```

and if the option ‘ER_FLAT’ is given the program does

```plaintext
if (ERf < a) then ERf = ER9; else ERf = b, ER9 < b < ER0
```

The cloud interface module can recognize ASCII data files of certain format, in particular the FLUENT-UNS profile file format. Geometrical transformations to align the FLUENT coordinate system with the one used in FLACS-CFD are available. These are:

- INIT: initialise the transformation matrix
- TRANSLATE tx,ty,tz
  
  tx, ty, tz translation vector
- SCALE tx,ty,tz, scale
  
  tx, ty, tz center of scaling
  scale scaling factor
- ROTATE tx,ty,tz, ax,ay,az, revolution, angle
  
  tx, ty, tz point on the rotation axis
  ax, ay, az rotation vector (direction of the rotation axis; must not be a zero vector)
  revolution unit rotation (e.g. 360 when using degrees)
  angle rotation angle (e.g. in degrees; the right hand rule yields the direction of the rotation based on the rotation vector)

Note:

In FLACS v10.1 and later the cloud file can be edited through the CASD scenario menu.
**4.7.1.8.1 Potential problems using the cloud interface**  
The cloud interface uses two parameters, \(\text{CV\_SIZE} \, [\text{m}]\) and \(\text{CV\_RESOLUTION} \, [-]\), to convert the gas concentration volumes specified in the cloud file to gas concentrations on the computational grid. The parameters are set, depending on the computational grid, to obtain a precise representation of the cloud.

When, in some regions or the domain, the grid is fine compared to the size of the cloud, the gas cloud mapping to the grid performed by the cloud interface during the simulation may take too long time.

To reduce the computation time, the parameters \(\text{CV\_SIZE}\) and \(\text{CV\_RESOLUTION}\) can be set in the cloud file, as seen in the example above (these parameters are not available in CASD).

In the example, \(\text{CV\_SIZE}\) is explicitly set to 1.0, overriding the value calculated by the cloud interface. When \(\text{CV\_SIZE}\) is explicitly set, it is recommended to choose a representative grid spacing \([\text{m}]\) of the region(s) where the fuel cloud(s) are defined.

Additionally, \(\text{CV\_RESOLUTION}\), which is the per control volume resolution (non-dimensional, natural number), is set to 3, overriding the default value of 10. As in this example, \(\text{CV\_RESOLUTION}\) can be set explicitly to a value lower than the default, to speedup the calculation of the fuel concentration field based on the input given in the \(\text{cs000000.CLOUD}\) file.

Whenever \(\text{CV\_SIZE}\) or \(\text{CV\_RESOLUTION}\) are set in the cloud file, overriding the defaults, the fuel cloud must be inspected (e.g. plotting the fuel mass fraction in Flowvis) to make sure there are no inaccuracies in local concentration values.

The result of the cloud mapping depends also on the number of control volumes discretizing the cloud and on the matching between grid lines and cloud enveloping surface. In general, the cloud interface does not guarantee that the original mass of fuel specified in the cloud file is preserved. Thus, when the grid resolution is coarse comparing to the size of the cloud, you may expect a small variation of the fuel mass.

**4.7.1.9 Events file**

An events file is an optional text file that is used to manipulate parameter settings of scenario objects like leaks and panels, while the simulator is running. Events files are automatically detected by the simulator if it exist in the scenario folder. The name of an events file has to match this template:

\(\text{cs000000.EVENTS}\)

The first line in an events file has to be:

```
VERSION 1.0
```

The file specifies a number of objects (e.g. timers, triggers or trigger_panel), parameter settings for triggering the event, and parameter settings that the event shall enforce on a given scenario object.

Each object in the events file should be given a name. The 3 first characters in the name are reserved for special internal use and should be set to "---".

A timer can be used to activate a panel or start a leak at a specified time. The following line will make panel #1 start to open at 0.460 sec, and it shall be fully opened after 0.040 sec:

```
timer(name="---MyTimer",time=0.460,panel=1,duration=0.040)
```

A trigger can be used to activate a panel or start a leak when the pressure in a monitor point exceed a specified value. The following line will start leak #1 0.002 sec after the pressure in monitor point 1 exceeds 0.050 barg, and it will stop 10 sec after it was started:

```
trigger(name="---MyTrigger",delay=0.002,leak=1,duration=10,monitor=1,p>0.050)
```

All leaks, panels, and monitor points used in the event file must be defined in the cs-file.

- A panel with "INITIAL\_AND\_FINAL\_POROSITY 0 1" will open when activated.
- A panel with "INITIAL\_AND\_FINAL\_POROSITY 1 0" will close when activated.
Other variables than pressure '$p>$' could be used to set the trigger. Combustion products '$prod>$' could for example be used as a flame detector, and set to trigger a suppression system or open a panel. The porosity '$por>$' of one panel can also be used to activate another panel. The following table gives an overview of the different object types, and which parameters each object type supports for triggering events and manipulating scenario objects.

Table 4.14: Event object types, and supported parameters

<table>
<thead>
<tr>
<th>Event objects</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>timer(...)</td>
<td>name=</td>
<td>object name, 3 first characters should be &quot;---&quot; by default</td>
</tr>
<tr>
<td></td>
<td>time=</td>
<td>activation time for this timer</td>
</tr>
<tr>
<td></td>
<td>panel=</td>
<td>panel number</td>
</tr>
<tr>
<td></td>
<td>leak=</td>
<td>leak number</td>
</tr>
<tr>
<td></td>
<td>opd=</td>
<td>panel opening pressure differences (optional)</td>
</tr>
<tr>
<td></td>
<td>duration=</td>
<td>duration for this timer (optional)</td>
</tr>
<tr>
<td>trigger(...)</td>
<td>name=</td>
<td>object name, 3 first characters should be &quot;---&quot; by default</td>
</tr>
<tr>
<td></td>
<td>delay=</td>
<td>delay time for this trigger</td>
</tr>
<tr>
<td></td>
<td>panel=</td>
<td>panel number</td>
</tr>
<tr>
<td></td>
<td>leak=</td>
<td>leak number</td>
</tr>
<tr>
<td></td>
<td>monitor=</td>
<td>monitor number</td>
</tr>
<tr>
<td></td>
<td>prod&gt;</td>
<td>trigger value for combustion products mass fraction</td>
</tr>
<tr>
<td></td>
<td>p&gt;</td>
<td>trigger value for pressure</td>
</tr>
<tr>
<td></td>
<td>T&gt;</td>
<td>trigger value for temperature</td>
</tr>
<tr>
<td></td>
<td>opd=</td>
<td>panel opening pressure differences (optional)</td>
</tr>
<tr>
<td></td>
<td>duration=</td>
<td>duration for this trigger (can be set to 0.0 for instantaneous change)</td>
</tr>
<tr>
<td>trigger_panel(...)</td>
<td>name=</td>
<td>object name, 3 first characters should be &quot;---&quot; by default</td>
</tr>
<tr>
<td></td>
<td>delay=</td>
<td>delay time for this trigger</td>
</tr>
<tr>
<td></td>
<td>panels=</td>
<td>panel numbers</td>
</tr>
<tr>
<td></td>
<td>por&gt;</td>
<td>trigger value for the panel-2 porosity</td>
</tr>
<tr>
<td></td>
<td>opd=</td>
<td>panel opening pressure differences (optional)</td>
</tr>
<tr>
<td></td>
<td>duration=</td>
<td>duration for this trigger (can be set to 0.0 for instantaneous change)</td>
</tr>
</tbody>
</table>

Note:
- Lines starting with "!" or "#" are taken as comments.
- The 'opd=' and 'duration=' setting should not be used together as they are mutually exclusive.
- When using the 'opd=' for a panel, the values will be applied when the event occurs.
- When using the 'duration=' for a panel, the panel will open/close during the given duration.
- When using the 'duration=' for a leak, the leak will be stopped after the given duration.
- Maximum number of event objects is 100.
4.7.1.10 Heat file

Optional file used to specify heat transfer by convection.

File name template: cs000000.HEAT

It is possible to input planes and boxes with given heat fluxes and temperatures. This interface is designed for testing purposes and is not yet very user-friendly. You enter data for the heat objects in a cs.HEAT file. Below is a description of the format.

Note:

Heat switch (HEAT_SWITCH) must be set to "1" for the heat file to be read.

```
VERSION 0.0
!-----------------------------------------------------------------------
! Using free format (20 fields separated by commas):
! 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,
! Field numbers and their content:
! 1 type,xP,yP,zP,xL,yL,zL,q,qx1,qx2,qy1,qy2,qz1,qz2,Tx1,Tx2,Ty1,Ty2,Tz1,Tz2,
! Empty fields are allowed:
! type,xP,yP,zP,xL,yL,zL, , , , , , , ,Tx1,Tx2,Ty1,Ty2,Tz1,Tz2,
! type,xP,yP,zP,xL,yL,zL,q,qx1,qx2,qy1,qy2,qz1,qz2, , , , , , ,
! Object types:
! type = type of heat object
! 001 : box
! 100 : heat on all solid surfaces
! 101 : heat on all solid surfaces inside box
! Units of measurements:
! [xyz] = Coordinates and sizes (m)
! q = Volume heat flux (J/m³/s)
! q[xyz] = Surface heat fluxes (J/m²/s)
! T[xyz] = Surface temperatures (degrees Celsius)
! [12] = Surface side (1=lower, 2=upper)
! Miscellaneous:
! first line gives file format version (currently 'VERSION 0.0')
! blank lines are *not* allowed in 'VERSION 0.0'
! '#' or '! ' in first column gives comment lines
! Examples:
! Box with constant temperature:
! 1, 0.,0.,0., 1.,1.,1.,, , , , , , , , 20.,20.,20.,20.,20.,20.,
! Box with constant surface heat fluxes:
! 1, 0.,0.,0., 1.,1.,1.,,10.,10.,10.,10.,10.,
! Box with constant volume heat flux:
! 1, 0.,0.,0., 1.,1.,1.,1.0E3, , , , , ,
!-----------------------------------------------------------------------
! ENTER YOUR HEAT OBJECTS HERE
# ...
```

For object with type 100, 'heat on all solid surfaces', the heat flux modelling specified for the object is applied to all solid surfaces within the computational domain (i.e. including all solid surfaces, if any, that are outside the box-shaped region as specified by the parameters 'xP,yP,zP,xL,yL,zL'). For object with type 101, 'heat on all solid surfaces inside box', the heat flux modelling specified for the object is only applied to solid surfaces within (inside or on the border of) the box-shaped region (not applied to any solid surfaces outside this box-shaped region). For object with type 001, 'box', the heat flux modelling specified for the
object is only applied to solid surfaces placed at the border (the six boundaries in x-, y-, z-direction) of the box-shaped region.
Convective heat transfer (when 'WALLF 1', 'HEAT_SWITCH 1', in the scenario file) is calculated in FLACS-CFD employing wall functions that model the detailed flow in the boundary-layer. A two-layer near-wall model for the momentum and thermal boundary-layer is implemented in the three-dimensional Control Volume (CV) scheme for the numerical solver in FLACS-CFD, reported in (Sand & Bakke, 1989). The implemented near-wall model has effect on the transport equations for momentum, enthalpy (energy), and turbulence (turbulent kinetic energy, and dissipation of turbulent kinetic energy). Note that wall functions with convective heat transfer are only modelled for CV faces of the numerical grid that are solid (corresponding to CV area porosity equal zero), and with adjacent CV volume regarded as open (corresponding to CV volume porosity equal one). Thus, for example, the surface of a cylindrical pipe will in general not be modelled with wall functions and convective heat transfer. If the geometry is complex within a certain region, e.g. containing groups of pipes, and the user has an estimate of the effective heat flux within this region, the user may consider to model volume heat flux (q ‘Volume heat flux (J/m3/s)’) corresponding to this effective heat flux.

To avoid any confusion, it is advised to not define for the same object both q[xyz] ‘Surface heat fluxes (J/m2/s)’ (specified by the parameters ‘qx1,qx2,qy1,qy2,qz1,qz2’), and T[xyz] ‘Surface temperatures (degrees Celsius)’ (specified by the parameters ‘Tx1,Tx2,Ty1,Ty2,Tz1,Tz2’). But if both are defined, convective heat transfer for solid surfaces will be modelled based on the temperature parameters ‘Tx1,Tx2,Ty1,Ty2,Tz1,Tz2’ (assuming that all the temperature parameters are for temperatures above the absolute zero; 0K equal -273.15 Celsius). Convective heat transfer for solid surfaces can optionally be combined for the same object with specification of q ‘Volume heat flux (J/m3/s)’. And an object can also be defined specifying only q ‘Volume heat flux (J/m3/s)’. The model for volume heat flux (specification of q ‘Volume heat flux (J/m3/s)’ for an object), does not assume that there are any solid surfaces within the box-shaped region (this region could be completely open).

If for the same object; empty fields are set for both the parameters ‘qx1,qx2,qy1,qy2,qz1,qz2’, and the parameters ‘Tx1,Tx2,Ty1,Ty2,Tz1,Tz2’, then no heat transfer will be modeled for the solid surfaces (completely, thermal insulation of surfaces).
When several objects are defined, it is advised that the box-shaped regions do not overlap, there is no automatic warning if overlapping regions (if overlapping regions, they will be read in the order added in the cs.HEAT file).

4.7.1.11 Pool setup section

The POOL scenario section is used to specify a liquid spill on the ground.

See also:

Please see the section on the pool model for details regarding the model, its applicability and the involved parameters.

It is possible to derive a non-spreading pool from the default moving spill model. For a moving spill, the shallow-water equations are solved in two dimensions on the ground. Heat from the ground, the flow above the pool/spill, and the radiation from the sun and/or fire determine the evaporation rate. The composition of the liquid equals that for the gas composition, see Gas composition and volume. Remember to set

\[ \text{EQUIVALENCE\_RATIOS\_E0\_ER9} 1\times10^{30}, 0.0 \]

Input variables are given as follows:

```
POOL
  STATIC_POOL 1
  POOL_SHAPE "CIRCULAR"
  START_POOL 5
  MASS_POOL_0 0
  DMDT 400
  POSITION 5 5 3
  RAD_I 0
```
The available parameters for a pool scenario are listed in the table below. Please be aware that the Pool section in CASD uses more readable names and descriptions for some of the parameters. In the Parameter column in the table below the names/descriptions used in CASD are written in parentheses after the name used by the simulator.

Table 4.15: Pool model parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALBEDO</td>
<td>The albedo of the pool material in the wavelength range of the incoming solar radiation.</td>
<td>[-]</td>
</tr>
<tr>
<td>C_POOL</td>
<td>Specific heat of the liquid pool material.</td>
<td>[J/(kg K)]</td>
</tr>
<tr>
<td>CFL_POOL</td>
<td>CFL number for the pool calculations.</td>
<td>[-]</td>
</tr>
<tr>
<td>CONDC_A</td>
<td>The thermal conductivity of air.</td>
<td>[W/(m K)]</td>
</tr>
<tr>
<td>(Air Thermal Conductivity)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONDC_S</td>
<td>Ground conductivity when POOL_GROUND is &quot;USER&quot;.</td>
<td>[W/(m K)]</td>
</tr>
<tr>
<td>(Ground thermal conductivity)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DADT</td>
<td>Area growth rate of the pool.</td>
<td>[m^2/s]</td>
</tr>
<tr>
<td>DATABASE</td>
<td>Vapour pressure relation coefficients to be used from FLACS-CFD database (&quot;YES&quot;) or manually specified.</td>
<td></td>
</tr>
<tr>
<td>(Use database)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DIFFUS_S</td>
<td>Ground thermal diffusivity when POOL_GROUND is &quot;USER&quot;.</td>
<td>[m^2/s]</td>
</tr>
<tr>
<td>DMDT</td>
<td>Mass rate inserted uniformly over the initial pool area.</td>
<td>[kg/s]</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Unit</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>EMISSION_A</td>
<td>Emission coefficient for air.</td>
<td>[-]</td>
</tr>
<tr>
<td>EMISSION_P</td>
<td>Emission coefficient for the pool, heat due to radiation.</td>
<td>[-]</td>
</tr>
<tr>
<td>H_WATER</td>
<td>Heat transfer coefficient between water and the pool.</td>
<td>[W/(m² K)]</td>
</tr>
<tr>
<td>HEAT_EVAP</td>
<td>Evaporation heat (for temperature at or close to normal boiling point) of evaporating liquid. When DATABASE is set to 'YES', HEAT_EVAP is also calculated later for each time step to reflect the dependency on the pool temperature.</td>
<td>[J/kg]</td>
</tr>
<tr>
<td>HEAT_SUN</td>
<td>(Heat sun) The radiative heat from the sun.</td>
<td>[W/m²]</td>
</tr>
<tr>
<td>KIN_VISC_A</td>
<td>Kinematic viscosity for air.</td>
<td>[m²/s]</td>
</tr>
<tr>
<td>MASS_POOL_0</td>
<td>(Mass) Pool mass at time START_POOL.</td>
<td>[kg]</td>
</tr>
<tr>
<td>MOL_WEIGHT</td>
<td>Molecular weight of the evaporating pool liquid.</td>
<td>[kg/kmol]</td>
</tr>
<tr>
<td>PRANDTL_A</td>
<td>Prandtl number of air.</td>
<td>[-]</td>
</tr>
<tr>
<td>PLATE_L</td>
<td>(Plate thickness) Thickness of the steel plate; must be defined when POOL_GROUND is &quot;PLATE&quot;.</td>
<td>[m]</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Unit</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>POOL GROUND</td>
<td>Possible grounds are CONCRETE, AVERAGE, WATER, SOIL, PLATE, INSULATED, and USER. If USER is specified, values for ground conductivity CONDUC_S and ground diffusivity DIFFUS_S must be added to the setup file. It is also possible to specify more than one ground property as follows: 1. Give the general ground at the beginning of the text string: GROUND1. 2. Specify the other ground and a box where this ground properties should be used: GROUND1,GROUND2 [x_low x_high y_low y_high z_low z_high]. Example: SOIL,WATER[: -1 0.1]. &quot;:&quot; means the entire range and can only be used for x and y directions. Conductivities and diffusivities for predefined ground are given in the Ground properties table.</td>
<td></td>
</tr>
<tr>
<td>POOL SHAPE</td>
<td>Currently only CIRCULAR pools are supported.</td>
<td></td>
</tr>
<tr>
<td>POSITION</td>
<td>The X, Y, and Z coordinates of the center of the pool (non-spreading pool) or leakage area (spreading pool). The Z coordinate is the altitude from where Flacs searches for the ground in the negative Z-direction. If Z is inside an object, there are two possible outcomes: 1. If there is no gap below Z, then Z = ZMAX. 2. If there is a gap (free space) below Z, then Z is the altitude of the surface below the gap. How to set the Z coordinate is emphasised in the Pool spread best practice example.</td>
<td>m</td>
</tr>
</tbody>
</table>
### 4.7 Files in FLACS-CFD

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>RAD_I (Inner radius)</td>
<td>Defines the inner radius of the donut shaped leakage area. RAD_I=0.0 is the default donut-shaped pool area. When STATIC_POOL=1, RAD_I specifies the inner radius of the donut-shaped pool area.</td>
<td>m</td>
</tr>
<tr>
<td>RAD_O (Outer radius)</td>
<td>The outer radius of the leak area. When STATIC_POOL=1, RAD_O specifies the outer radius of the pool area.</td>
<td>m</td>
</tr>
<tr>
<td>RAD_T</td>
<td>The start radius of the liquid pool.</td>
<td>m</td>
</tr>
<tr>
<td>ROUGH_L (Surface roughness)</td>
<td>Surface roughness over the pool, default value is 0.005 m.</td>
<td>m</td>
</tr>
<tr>
<td>START_POOL (Start time)</td>
<td>Time when pool leakage starts and the pool starts to evaporate.</td>
<td>s</td>
</tr>
<tr>
<td>STATIC_POOL (Non-spreading pool)</td>
<td>The default STATIC_POOL=0 refers to a moving (dynamic) spill, where heat and mass transfer are calculated locally for each control volume. Set STATIC_POOL=1 to deactivate spreading (to model a static circular pool with macroscopic correlations for the heat and mass transfer).</td>
<td></td>
</tr>
<tr>
<td>T_BOIL</td>
<td>Temperature at the boiling point of the evaporating liquid.</td>
<td>K</td>
</tr>
<tr>
<td>T_POOL</td>
<td>Temperature of the pool (uniform value updated each time step).</td>
<td>K</td>
</tr>
<tr>
<td>T_SOIL (Ground temperature)</td>
<td>The ground temperature.</td>
<td>K</td>
</tr>
<tr>
<td>VEL_WATER (Water velocity)</td>
<td>Two-dimensional vector (vel_x, vel_y) describing the velocity of water, default is 0.0,0.0.</td>
<td>m/s</td>
</tr>
</tbody>
</table>
| POOL_USE_RADIATION    | When running a coupled pool-fire scenario four different settings may be applied:  
  - 0 : use zero radiation from the flame  
  - 1 : use a fixed radiation level  
  - 2 : use “ambient” radiation (for backward compatibility), i.e. radiation calculated from the surrounding temperature  
  - 3 : use the radiation field from the radiation model. | [-] Only for fire |
<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>POOL_FIXED_RADIATION</td>
<td>The level of radiative heat transferred to the pool surface is possible to control through this control variable. This variable is employed for POOL_USE_RADIATION=1 and POOL_USE_RADIATION=3 (if no radiation model is chosen), the default value is 10000.</td>
<td>[W/m$^2$] Only for fire</td>
</tr>
<tr>
<td>POOL_MINIMUM_RADIATION</td>
<td>The minimum level of radiative heat to the pool surface is possible to control through this control variable. This control variable is employed for POOL_USE_RADIATION=2 and POOL_USE_RADIATION=3, the default value is 10000.</td>
<td>[W/m$^2$] Only for fire</td>
</tr>
<tr>
<td>POOL_USE_TURBULENCE</td>
<td>Turbulence above pool surface:</td>
<td>[-] Only for fire</td>
</tr>
<tr>
<td></td>
<td>• 0 : do not apply any turbulence modifications</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 1 : use $u_{ref}$ = velocity of vapor entering from the pool, set TKE = $1.5 \times (RTI \times u_{ref})^{\frac{3}{2}}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 2 : use $u_{ref}$ = velocity of vapor entering from the pool, set TKE = $\max(TKE, 1.5 \times (RTI \times u_{ref})^{\frac{3}{2}})$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 3 : use $u_{ref}$ = velocity of bulk flow just above the pool, set TKE = $1.5 \times (RTI \times u_{ref})^{\frac{3}{2}}$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 4 : use $u_{ref}$ = velocity of bulk flow just above the pool, set TKE = $\max(TKE, 1.5 \times (RTI \times u_{ref})^{\frac{3}{2}})$.</td>
<td></td>
</tr>
<tr>
<td>POOL_RTI</td>
<td>If POOL_RTI &gt; 0 then set TKE accordingly just above the pool.</td>
<td>[-] Only for fire</td>
</tr>
<tr>
<td>POOL_TLS</td>
<td>If POOL_TLS &gt; 0 then set TLS accordingly just above the pool.</td>
<td>[m] Only for fire</td>
</tr>
</tbody>
</table>

Table 4.16: Ground properties

<table>
<thead>
<tr>
<th>Ground</th>
<th>Conductivity (Wm$^{-1}$K$^{-1}$)</th>
<th>Thermal diffusivity (m$^2$s$^{-1}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concrete</td>
<td>1.1</td>
<td>$10^{-6}$</td>
</tr>
<tr>
<td>Average/Soil</td>
<td>0.9</td>
<td>$4.3 \times 10^{-7}$</td>
</tr>
<tr>
<td>Plate</td>
<td>15</td>
<td>$3.9 \times 10^{-6}$</td>
</tr>
<tr>
<td>Insulated</td>
<td>0.0</td>
<td>$10^{30}$</td>
</tr>
<tr>
<td>User</td>
<td>Set CONDUC_S</td>
<td>Set DIFFUS_S</td>
</tr>
<tr>
<td>Water</td>
<td>Heat transfer coefficient</td>
<td>Heat transfer coefficient</td>
</tr>
</tbody>
</table>
4.7 Files in FLACS-CFD

See also:
Example file included with the FLACS-CFD installation. Linux:

> cp /usr/local/Gexcon/FLACS-CFD_22.2/doc/examples/ex02_pool/cs600000.dat3

Windows: Copy files from

C:\Program Files\Gexcon\FLACS-CFD_22.2\doc\examples\ex02_pool\cs600000.dat3

4.7.1.12 Pool leakage file

This is an optional file and it requires that there is a POOL section in the cs-file and that RunManager → Parameters → FLACS-CFD version = pool.

File name template: cl000000.POOL

The cl-file contains:

- Line 1: 'POOL LEAK FILE'
- Line 2: Number of variables
- Line 3: Name of columns. 'TIME' should always be in the first column.
- Line 4 - : Values at different points in time.

Example:

```
'POOL LEAK FILE'
2
'TIME [s]' 'DMDT [kg/s]'
0.0 0.0
1.0 1000.0
100.0 1000.0
101.0 0.0
1200 0.0
```

Attention:

Time in the cl000000.POOL is relative to START_POOL in the POOL section in cs000000.dat3.

See also:
Example file included with the FLACS-CFD installation. Linux:

> cp /usr/local/Gexcon/FLACS-CFD_22.2/doc/examples/ex02_pool/cl600000.POOL .

Windows: Copy files from

C:\Program Files\Gexcon\FLACS-CFD_22.2\doc\examples\ex02_pool\cl600000.POOL

4.7.1.13 Leak file

Leak files define time-dependent leaks of either fuel or suppressant. Leak files may be generated manually, or by using the jet utility program. File name template: c1000000.n001 The index number in the file name extension identifies the leak.
See also:

Leaks section in the Technical Reference chapter and the Leaks section in the CASD chapter.

For brevity, these files may be called cl-files or leak files hereafter. They are text files and may be generated using a text editor or using CASD.

It is possible to simulate realistic leaks in FLACS-CFD. This is done by proper sub-grid modelling of the conditions at the leak outlet and by allowing time varying leak data and multiple leaks. The position and direction vector for each leak must be specified in the scenario file. For each leak there has to be a cl-file containing the time dependent leak data. Depending on the leak specification in the scenario file, the Flacs simulator may auto-generate this file before running the simulation.

The leak number is used as identification number in the file type.

The format of the cl-file is as follows:

```
LEAK_CONTROL_STRING
7
TIME AREA RATE VEL RTI TLS T
```

The seven parameters that constitute the time dependent leak data are chosen so that it is possible to specify rate, velocity, turbulence, and temperature for the leak. (at the nozzle throat) The names in the file format described above have the following meaning:

**Table 4.17: Leak parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TIME</td>
<td>(s)</td>
<td>Time</td>
</tr>
<tr>
<td>AREA</td>
<td>(m²)</td>
<td>Area</td>
</tr>
<tr>
<td>RATE</td>
<td>(kg/s)</td>
<td>Mass flow</td>
</tr>
<tr>
<td>VEL</td>
<td>(m/s)</td>
<td>Velocity</td>
</tr>
<tr>
<td>RTI</td>
<td>(-)</td>
<td>Relative turbulence intensity</td>
</tr>
<tr>
<td>TLS</td>
<td>(m)</td>
<td>Turbulence length scale</td>
</tr>
<tr>
<td>T</td>
<td>(K)</td>
<td>Temperature</td>
</tr>
</tbody>
</table>

Details of the calculation of turbulent kinetic energy and its rate of dissipation from the set of parameters listed above (VEL, RTI and TLS) are given for the WIND condition.

Line number one of the leak file contains the LEAK_CONTROL_STRING which is used to specify the type of leak and the leak outlet. The following is recognised:

**Table 4.18: Leak control string letters**

<table>
<thead>
<tr>
<th>Letter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>Jet leak</td>
</tr>
<tr>
<td>D</td>
<td>Diffuse leak</td>
</tr>
<tr>
<td>E</td>
<td>Jet leak with modelled entrainment</td>
</tr>
<tr>
<td>+</td>
<td>Direction is now positive</td>
</tr>
<tr>
<td>-</td>
<td>Direction is now negative</td>
</tr>
<tr>
<td>X</td>
<td>Leak through control volume face perpendicular to the x-axis</td>
</tr>
<tr>
<td>Y</td>
<td>Leak through control volume face perpendicular to the y-axis</td>
</tr>
<tr>
<td>Z</td>
<td>Leak through control volume face perpendicular to the z-axis</td>
</tr>
<tr>
<td>:lean</td>
<td>Air leak (same as :mix 0)</td>
</tr>
<tr>
<td>:rich</td>
<td>Fuel leak (default, same as :mix 1)</td>
</tr>
</tbody>
</table>
Diffuse leaks are handled by setting fixed values for the scalar quantities and introducing a mass source equal to the specified leak rate at the leak node. The area porosity at the leak outlet is set so that the specified area is obtained. A leak velocity that satisfies mass conservation is obtained automatically through the flow field solution procedure.

Jet leaks are handled by setting fixed values for the scalar quantities and one or more of the momentum components at the leak node. The area porosity is calculated so that the specified mass flux and momentum flux is obtained. The mass conservation is not solved for at a jet leak node during the flow field calculation. The default leak type is diffuse (D) and the default direction is positive (+). Take care not to specify a control volume face more than once. The following examples deal with the LEAK_CONTROL_STRING.

If a FAN with direction +X is specified, this will be translated into the string ”!J+X=X:fan” in the leak file. Fans should be along the axis directions. To change the mixture fraction of the leak use :mix and the mixture fraction, e.g. ”J+X:mix 0.1” gives a leak in direction X with a mixture fraction of 0.1. The ‘!’ prefix in the leak string is used to allow the Flacs simulator to overwrite the leak file. When manually creating or editing the cl-file outside CASD, or using an external script to generate it, there should be no ‘!’.

Attention:
Opening a scenario with existing, write-protected cl-files (removed ‘!’) in CASD will not keep the cl-file(s) unchanged. Manually edited specifications like ‘=’, ‘:fan’ or ‘reversed’ may be removed, changing the character of the leak or leading to conflicting information in the cs- and cl-files, cf. the warning about limited support of leak files in CASD.

4.7.1.13.1 How to avoid automatic overwriting of leak files by Flacs
It is possible to prevent the automatic overwriting of an existing leak file by the Flacs simulator at the beginning of the simulation: Modify the leak file (e.g. cl654321.n001 for the first leak of job number 654321) and remove the character ‘!’ from the first line. For example replace
‘!J+Y’
with
‘J+Y’
This will ensure that the Flacs simulator does not write a new leak file at the beginning of the simulation.

4.7.1.13.2 Example 1: Diffuse leak
‘D+X’

4.7.1.13.3 Example 2: Jet leak in positive x-direction:
‘J+X’

4.7.1.13.4 Example 3: Setting the mixture fraction for leak:
‘J+X:mix 0.1’
### 4.7.1.13.5 Example 4: Erroneous specification:

'\(DJ+X+XY\)'

The following error messages would be issued on the log file(s):

```plaintext
*** (LKINIT) ALREADY SET DIFFUSE
*** (LKINIT) ALREADY SET +X
```

The simulation does not stop unless severe errors occur. In this example, the leak specification would be taken as:

'\(D+XY\)'

The `LEAK_CONTROL_STRING` must be enclosed in single quotes (‘...’) and may contain blanks. The maximum length is 16 characters.

* Line number two contains the number of data values to be found in each of the next lines.
* Line number three contains the names of the variables. The sequence of the variables is predefined and may not be changed. The time will always be the first value on a line containing data values.
* Line number four and onward contain values for each variable. As the simulation proceeds the file will be scanned for new data values when old values expire. Intermediate values are calculated as time averages of the inputs from the cn-file between the old time and the new time.
* A leak is started when the first time on its leak file is reached and is stopped when the last time on the file is reached.

In the case of a jet leak, the nozzle density, `DENS`, and pressure, `PRES`, are calculated as follows:

\[
DENS = RATE / (VEL * AREA) \\
PRES = DENS * R * T
\]

Where \( R \) is the specific ideal gas constant.

If the nozzle pressure is higher than the ambient pressure an analytic jet expansion will be calculated.

The gas composition for each leak is taken to be the one specified by the equivalence ratio for rich gas, `ER0`, and volume fractions given on the scenario file.

The leak position and direction vector is given in the scenario file. The direction vector does not need to be of unit length. FLACS-CFD will calculate the direction cosines by dividing each component in the direction vector by the length of the vector. The direction cosines are used to modify the flow velocity and the area for the X-, Y- and Z-direction independently. This may be used to simulate jet leaks that are not parallel to the main axes. It is noted that this option should be used with care in cases where the grid resolution is coarse due to the fact that numerical diffusion tends to smear out the jet.

### 4.7.1.13.6 Example 5: Jet at 45.0 degrees:

'\(J+XY\)'

Direction vector given as 1.0 1.0 0.0

### 4.7.1.13.7 Example 6: Jet at 0.0 degrees:

'\(J+X\)'

Direction vector given as 1.0 0.0 0.0

### 4.7.1.13.8 Example 7: Elliptic area leak with a Gaussian release profile in the upward direction:

'\(J+Z:AreaLeak(profile=Gaussian,shape=Elliptical)\)'

---

FLACS-CFD v22.2 User's Manual
4.7 Files in FLACS-CFD

4.7.1.13.9 Example 8: Suction rectangular area leak with a uniform profile; the following specification leads to a flow towards the suction location (oriented towards the positive x-direction), and the flow disappears from the domain at the suction location:

\['J-X:AreaLeak\{profile=\text{Uniform},\ shape=\text{Rectangular}\}:\text{reverse}\]’

4.7.1.14 Runtime simulation control file

Optional file, can be used to define output, generate or load dump files, terminate the simulation, etc.

File name template: cc000000.dat3

For brevity this file may be called the cc-file or control file hereafter. It is a text file and may be generated using a text editor. The cc-file is a facility for giving instructions to FLACS-CFD during a simulation. Keywords and values may be given on the cc-file which will be read and interpreted each time step. It may contain any number of lines. Each line shall contain a keyword followed by a value separated by a comma (,) or a space ( ). Float values may be entered with or without a decimal point, integer values are taken to be the integer part of a float value. The following keywords are recognised:

Table 4.19: Valid cc-file keywords

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>NLOAD</td>
<td>(deprecated, please use LOAD in the cs-file instead) Set load identification and load data at simulation start</td>
</tr>
<tr>
<td>NDUMP</td>
<td>Set identification number for data dump (8887 and 8888 are reserved dumpids and en error is raised if NDUMP is set to either of these)</td>
</tr>
<tr>
<td>TDUMP</td>
<td>Dump data at given time</td>
</tr>
<tr>
<td>IDUMP</td>
<td>Dump data at given iteration number</td>
</tr>
<tr>
<td>FDUMP</td>
<td>Dump data at given fuel level</td>
</tr>
<tr>
<td>TOUTF</td>
<td>Write results at given time</td>
</tr>
<tr>
<td>IOUTF</td>
<td>Write results at given iteration number</td>
</tr>
<tr>
<td>FOUTF</td>
<td>Write results at given fuel level</td>
</tr>
<tr>
<td>TSTOP</td>
<td>Stop simulation at given time</td>
</tr>
<tr>
<td>ISTOP</td>
<td>Stop simulation at given iteration number</td>
</tr>
<tr>
<td>FSTOP</td>
<td>Stop simulation at given fuel level</td>
</tr>
</tbody>
</table>

Fuel level is defined as the current total mass of fuel divided by the initial total mass of fuel. The fuel level cannot be used for gas dispersion simulations where the initial fuel mass is zero.

The control file is useful when you want to produce output that is not possible to specify on the scenario file and when you want to monitor a simulation while it is running. The dump option is one example of output and input that is not possible to specify on the scenario file. The following examples shows how to use the cc-file.

Note:

cc-file settings can also be specified in the Dump/load settings dialogue in CASD.

4.7.1.14.1 Example 1 Contents of cc-file (including comments):

NDUMP 1 dump file name is "rd000000.n001"
TDUMP 0.20 dump when time is 0.2 s
NDUMP 2 dump file name is "rd000000.n002"
TDUMP 0.50 dump when time is 0.5 s
TDUMP 0.70 dump when time is 0.7 s

What happened:
1. The file "rd000000.n001" was written at time 0.20 s.
2. The file "rd000000.n002" was written at time 0.50 s.
3. The file "rd000000.n002" was written at time 0.70 s.

4.7.1.15 Time dependent CFL file

Optional file used to specify time dependent CFL numbers (see sections CFLC and CFLV for details).
File name template: cn000000.dat3
For brevity this file may be called the cn-file or CFL-number file hereafter. The cn-file is a facility for
 giving instructions to FLACS-CFD during a simulation.

Warning:

It is not advised to change the recommended and default time step settings in FLACS-CFD. This facility
should only be used for special cases and by experienced users who understand the consequences and
limitations.

High flow velocities and change rates and small control volumes put severe constraints on the size of
the time step in computational fluid dynamics. Using large time steps may cause the solution to become
unstable. However, to be able to simulate over a long period with today's computer resources, it is necessary
to maintain as large a time step as possible. During a gas dispersion simulation the stability requirements
vary, characterised by stronger requirements in periods of highly unsteady flows and weaker requirements in
steady flow periods.

To effectively simulate gas dispersion scenarios including high momentum jets, a facility for specifying time
dependent Courant numbers is available. FLACS-CFD will read time-varying Courant numbers from the
cn-file if it exists. Courant numbers specified on the scenario file are then overridden.

The cn-file is a text file whose format is shown below:

```
HEADLINE
3
TIME CFLC CFLV
data (1-3)
... ...
```

The first line is a comment for user information and not evaluated by the simulator. The second line contains
the number of data values to be found in each of the following lines, in this case the number of data values
is 3. The third line defines the names of the variables. The sequence of the variables in a cn-file is fixed and
may not be changed. The time will always be the first value on a line containing data values. Line number
four and onward contain values for each variable. As the simulation proceeds the file will be scanned for
new data values when old values expire. Intermediate values are calculated as time averages of the inputs
from the cn-file between the old time and the new time.

The cn-file can be generated using a text editor or by defining the necessary values in the Time dependent CFL
numbers section of the scenario menu in CASD.

4.7.1.16 Monitor file

If more than one monitor region is needed, then you must create a cs.MON file (e.g. cs010100.MON). This
file can also be used if more detailed output is needed.

The cs.MON file is a text file and can be edited with any text editor; it will automatically be read by
FLACS-CFD when the simulation of the scenario 123456 is started. The file may contain comment lines
starting with '!' or '#', but must start with VERSION 1.0. The other content should specify monitor setups
with the parameters listed in the following table.
### Table 4.20: Monitor file parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>&lt;suffix&gt;</td>
<td>unique name of the monitor object; max 32 characters; its output file (if generated) will be named rt010100.MON.&lt;suffix&gt;</td>
</tr>
<tr>
<td>start</td>
<td>x,y,z</td>
<td>start position of the considered region in space, unit [m]</td>
</tr>
<tr>
<td>end</td>
<td>x,y,z</td>
<td>end position of the considered region in space, unit [m]</td>
</tr>
<tr>
<td>see (only for type volume)</td>
<td>1.0</td>
<td>this parameter is set to a fixed value 1.0</td>
</tr>
<tr>
<td>mix (only for type volume)</td>
<td>&lt;fraction&gt;</td>
<td>when the output in the file rt010100.MON.&lt;suffix&gt; is calculated, the fuel concentration in every control volume is evaluated as the fuel concentration of standard FLACS-CFD multiplied with the factor &lt;fraction&gt;; used by the frozen cloud approach</td>
</tr>
<tr>
<td>output (only for type line)</td>
<td>&quot;string&quot;</td>
<td>output specification, max 128 characters; see section Line monitor output specification below</td>
</tr>
<tr>
<td>step (only for type line)</td>
<td>step_size</td>
<td>optional, can be used instead of steps=</td>
</tr>
<tr>
<td>steps (only for type line)</td>
<td>number_of_steps</td>
<td>optional, default steps=100</td>
</tr>
</tbody>
</table>

An example of a valid cs.MON file is:

```plaintext
VERSION 1.0
# This is a comment.
! Another comment line.
volume(name="123455",start=138,-21,32,end=198,21,48,
  output="FUEL(see=1.0,mix=0.8)")
volume(name="123456",start=138,-21,32,end=198,21,48,
  output="FUEL(see=1.0,mix=1)")
volume(name="123457",start=138,-21,32,end=198,21,48,
  output="FUEL(see=1.0,mix=1.2)")
line(name="LineMonitor1",output="FUEL(ER),details=1",start=142,0,36,
  end=158,0,40,steps=50)
line(name="PipeSeg1",output="PRESSURE(),details=1",start=150,8,42,
  end=170,8,42,step=0.2)
! More objects may be entered (up to the maximum of 100), one per line.
! The maximum length of the input text lines is 256 characters.
```

Note that the input lines are split for readability here in the manual, every volume or line entry must appear on a single line.

See also:

The section on monitor output files discusses examples.

#### 4.7.1.16.1 Line monitor output specification

The output specification may be one of the following:

- `output="FUEL()"` to get the fuel concentration (various options)
- `output="PRESSURE()"` to get the overpressure (average, [barg])

To get detailed output (for each of the points along the line) enter the following:

- `output="FUEL(),details=1"`
- `output="PRESSURE(),details=1"`

For the fuel output there are more options:
Table 4.21: Fuel output options for line monitors.

<table>
<thead>
<tr>
<th>Value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{FUEL()}$, or $\text{FUEL(MoleFraction)}$</td>
<td>$\int_{\text{start}}^{\text{end}} \text{FMOLE} , ds$</td>
</tr>
<tr>
<td>$\text{FUEL(MassFraction)}$</td>
<td>$\int_{\text{start}}^{\text{end}} \text{FUEL} , ds$</td>
</tr>
<tr>
<td>$\text{FUEL(ER)}$</td>
<td>$\int_{\text{start}}^{\text{end}} \text{ER} , ds$</td>
</tr>
<tr>
<td>$\text{FUEL(LFL)}$, or $\text{FUEL(LFL,MoleFraction)}$</td>
<td>$\int_{\text{start}}^{\text{end}} \frac{\text{FMOLE}}{\text{LFL}_{\text{MOL}}} , ds$</td>
</tr>
<tr>
<td>$\text{FUEL(LFL,MassFraction)}$</td>
<td>$\int_{\text{start}}^{\text{end}} \frac{\text{FUEL}}{\text{LFL}_{\text{FUEL}}} , ds$</td>
</tr>
<tr>
<td>$\text{FUEL(LFL, ER)}$</td>
<td>$\int_{\text{start}}^{\text{end}} \frac{\text{ER}}{\text{LFL}_{\text{ER}}} , ds$</td>
</tr>
</tbody>
</table>

By default a line monitor will use the LFL for the given mixture, however, it is possible to set the LFL value to be used by the line monitor:

- $\text{FUEL(LFL=0.5)}$ set LFL based on LFLe
- $\text{FUEL(ER,LFL=0.5)}$ set LFL based on LFLe
- $\text{FUEL(MoleLFL=0.049768)}$ set LFL based on LFLx
- $\text{FUEL(MassLFL=0.028302)}$ set LFL based on LFLy

The use of the different parameters and settings is demonstrated in the following sample cs.MON file.

```mon
VERSION 1.0
!
! Detailed FUEL results as [MoleFraction], [MassFraction] and [ER]:
! (for the given mixture and for a specified value) and average:
line(name="A1",start=0,0,0,end=30,20,10,output="FUEL(MoleFraction),details=1")
line(name="A2",start=0,0,0,end=30,20,10, output="FUEL(MassFraction),details=1")
line(name="A3",start=0,0,0,end=30,20,10, output="FUEL(ER),details=1")
!
! Integral FUEL results as [MoleFraction*m], [MoleFraction*m]/[MoleFraction at LFL]
line(name="F1",start=0,0,0,end=30,20,10, output="FUEL()")
line(name="F2",start=0,0,0,end=30,20,10, output="FUEL(LFL)")
line(name="F3",start=0,0,0,end=30,20,10, output="FUEL(MoleLFL=0.05)")
line(name="F4",start=0,0,0,end=30,20,10, output="FUEL(Average)")
!
! Average pressure and details as [Pa]:
line(name="P1",start=0,0,0,end=30,20,10, output="PRESSURE(),details=1")
!
! Average pressure and details as [Pa] use 200 steps along the line:
line(name="P2",start=0,0,0,end=30,20,10,steps=200, output="PRESSURE(),details=1")
```

4.7.2 Output files from FLACS-CFD simulations

This section summarises the various output files from FLACS-CFD simulations.

4.7.2.1 CGNS files

The simulator outputs data in the CGNS format. The main output file `<JOB>.cgns` stores or links to the following data:

- dump snapshots (content of Simulation dump file (Flacs2 only) in Flacs2);
- 3D results (content of Field output file (Flacs2 only) in Flacs2);
4.7 Files in FLACS-CFD

- 1D (scalar, time series) results (content of Scalar-time output file (Flacs2 only) in Flacs2);
- Other time series results, previously stored in multiple log files in Flacs2.

The CGNS file or files are self-contained and may contain links only to other CGNS pr HDF5 files. This means that it is not necessary to have any of the simulation setup files present in order to open the CGNS file in Flowvis or process it using the Python API. Note that for geometry visualisation and accurate interpolation the geometry and porosity files are still needed.

The 3D field output written to the CGNS file by the simulator follows the CGNS standard. The structure of the file is defined by the Standard Interface Data Structures (SIDS) definitions (CGNS UG).

The CGNS file can be viewed and manipulated using any of the tools described in CGNS tools.

See also:
A User's Guide to CGNS

Note:
Renaming the CGNS file also requires renaming the porosity and geometry files in order to open the simulation correctly in Flowvis. E.g., if a CGNS file is renamed as BigLeak.cgns, then the corresponding porosity file should be renamed to cpBigLeak.dat3, the co-file should be renamed as coBigLeak.geo (or coBigLeak.dat3 if the old CO file format is used).

Since version 22.2 of FLACS-CFD user can specify scenario menu item SPLIT,CGNS. By default, splitting is enabled for all 1D, 3D and dump output. When splitting is enabled, the main CGNS file <JOB>.cgns contains links to many smaller, incrementally written output files in the <JOB>.output/ directory - primary, chunk and grid files. The default setting addresses most of the issues common for large binary files, i.e.: risk of data corruption, incremental downloads, and ease of archiving.

There are 4 file types found in the simulation directory:

- main file (pattern <JOB>.cgns) - equivalent to the non-split CGNS file, but smaller in size, often updated in the course of simulation.
- primary files:
  - primary 1D file - <JOB>.output/<JOB>.1d.cgns - stores all 1D results.
  - 3D - <JOB>.output/<JOB>.3d.cgns - pointer file to 3D data (of no practical importance to the user).
  - dump - <JOB>.output/<JOB>.dump.cgns - pointer file to dump data (of no practical importance to the user).
- chunk files:
  - 3D chunk files (pattern <JOB>.output/<JOB>.3d<ITER>.cgns) - multiple self-contained CGNS files storing 3D data from one or more time steps flushed at the iteration number <ITER>. Linked only to the grid files. Can be previewed in Flowvis. Immutable in the course of simulation.
  - dump chunk files - <JOB>.output/<JOB>.dump<ITER>.cgns - multiple standalone CGNS files storing dump data at the iteration <ITER>. They may be archived and used to start simulations from dump, not linked to any external data.
- grid files (pattern <JOB>.output/<JOB>.grid.*.hdf5) - immutable in the course of simulation, of no practical importance to the user.

The main CGNS file contains links to the data stored in primary, chunk and grid files. Therefore, the postprocessing workflow has not changed and the main file has the same name and can be used in Flowvis the same way as the single non-split CGNS files.

In case of the dump files, these are supported workflows:
1. Suspend/resume using the main file
2. Loading dump file using the main file
3. Initialization from JLOAD file (e.g. multiple gas explosion simulations after dispersion)
   (a) using main CGNS file as a source
   (b) using archived dump chunk files as a source (new in 22.2) - chunk dump files are self-contained, therefore can be used as a staring point for the new simulation. User must rename the file to follow the pattern `<JLOAD>.cgns`.

4.7.2.1.1 Writing frequency

Due to performance reasons, simulator can postpone writing 1D and 3D results to the disk and instead buffer results in the memory and flush data to the CGNS file once buffers are filled.

1D data are silently written to the buffer every MODD iterations. Whenever 3D data are written to the buffer, simulator will write the following to the log file:

```
# writing to 3D results buffer at fixed time interval ...
```

Whenever 1D or 3D data are flushed to the disk, simulator will log the following respectively:

```
# flushing buffered 1D results to file
# flushing buffered 3D results to file
```

Additionally, unless the buffer is filled earlier, 1D data are flushed to the disk according to the following rules:

- not earlier than 30 seconds after previous flush,
- not later than 30 minutes after previous flush.

In case of the 3D data, the following rule applies additionally, unless the buffer is filled earlier:

- not later than 60 seconds after previous flush.

4.7.2.1.2 Getting time-series from CGNS file

All data stored in the CGNS file can be visualized in Flowvis. For advanced users the data is also available through the Python API. Some of the methods that can be used to access data are presented below.

See also:

See the Python API section for more information on the Python interface.

The time series of TIME variable can be obtained with the following script:

```python
import flacs.data
dataset = flacs.data.load_dataset("000000.cgns")
time = dataset.get_data(dataset.variable("TIME"),[[]]).data
```

All monitor point variables can be accessed either by the name or the index of the monitor point and the variable SIDS identifier as shown in the example below:

```python
import flacs.data
dataset = flacs.data.load_dataset("000000.cgns")
data1 = dataset.get_data_from_monitor_point_by_name("Pressure","MP 89").data
data2 = dataset.get_data_from_monitor_point_by_idx("Pressure",89).data
```

SIDS data-name identifiers that can be used as the first argument are listed in the table in Output variables in FLACS-CFD.

All monitor panel variables can be accessed either by the name or the index of the monitor panel and the variable SIDS identifier as shown in the example below:

```python
import flacs.data
dataset = flacs.data.load_dataset("000000.cgns")
data1 = dataset.get_data_from_pressure_panel_by_name("PanelPressure","Panel 1").data
data2 = dataset.get_data_from_pressure_panel_by_idx("PanelPressure",1).data
```

SIDS data-name identifiers that can be used as the first argument are listed in the table in Output variables in FLACS-CFD. In addition the following parameters are available:
Table 4.22: SIDS data-name identifiers of monitor panel time independent parameters

<table>
<thead>
<tr>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters/cd</td>
<td>(-)</td>
<td></td>
</tr>
<tr>
<td>Parameters/opendp</td>
<td>(-)</td>
<td>Opening pressure differences</td>
</tr>
<tr>
<td>Parameters/poros</td>
<td>(-)</td>
<td>Panel porosity</td>
</tr>
<tr>
<td>Parameters/subsize</td>
<td>(-)</td>
<td>Panel subsize</td>
</tr>
<tr>
<td>Parameters/tramax</td>
<td>(m)</td>
<td>Maximum travel distance</td>
</tr>
<tr>
<td>Parameters/weight</td>
<td>(kg)</td>
<td>Panel weight</td>
</tr>
</tbody>
</table>

Monitor line variable can be accessed by the name of the monitor line and the variable SIDS identifier Integral as shown in the example below:

```python
import flacs.data
dataset = flacs.data.load_dataset("000000.cgns")
data = dataset.get_data_from_pressure_panel_by_name("Integral","Panel 1").data
```

The unit and the meaning of the variable depends on the input, as described in Line monitor output specification.

All monitor region variables can be accessed by the name of the monitor region and the variable SIDS identifier as shown in the example below:

```python
import flacs.data
dataset = flacs.data.load_dataset("000000.cgns")
data1 = dataset.get_data_from_monitor_region_by_name("Integral","Default").data
data2 = dataset.get_data_from_monitor_region_by_name("Integral","MonitorRegion1").data
```

The Default monitor region name is reserved for the main gas monitor region (in Flacs2 reported to Fuel log file), all other names refer to Custom gas monitor regions.

SIDS data-name identifiers that can be used as the first argument are listed in tables below for monitor regions monitoring equivalent gas clouds and LFL percentiles.

Table 4.23: SIDS data-name identifiers of monitor region variables monitoring equivalent gas cloud

<table>
<thead>
<tr>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERfacMass</td>
<td>(kg)</td>
<td>reactivity scaled flammable mass, equivalent to Q1</td>
</tr>
<tr>
<td>FlamMass</td>
<td>(kg)</td>
<td>mass of the fuel composition (e.g. natural gas) where the concentration of fuel (when fuel is mixed with air) is within the flammable limits</td>
</tr>
<tr>
<td>FuelMass</td>
<td>(kg)</td>
<td>total fuel mass in the mixture</td>
</tr>
<tr>
<td>GasMass</td>
<td>(kg)</td>
<td>total gas mass in the mixture</td>
</tr>
<tr>
<td>FlamVol</td>
<td>(m3)</td>
<td>volume of the fuel-air mixture where the fuel concentration is within the flammable range</td>
</tr>
<tr>
<td>Q1</td>
<td>(kg)</td>
<td>equivalent cloud variable, see Gas monitor region</td>
</tr>
<tr>
<td>Q2</td>
<td>(kg)</td>
<td>equivalent cloud variable, see Gas monitor region</td>
</tr>
<tr>
<td>Q3</td>
<td>(kg)</td>
<td>equivalent cloud variable, see Gas monitor region</td>
</tr>
<tr>
<td>Q4</td>
<td>(m3)</td>
<td>equivalent cloud variable, see Gas monitor region</td>
</tr>
<tr>
<td>Q5</td>
<td>(m3)</td>
<td>equivalent cloud variable, see Gas monitor region</td>
</tr>
<tr>
<td>Q6</td>
<td>(m3)</td>
<td>equivalent cloud variable, see Gas monitor region</td>
</tr>
<tr>
<td>Q7</td>
<td>(m3)</td>
<td>equivalent cloud variable, see Gas monitor region</td>
</tr>
<tr>
<td>Q8</td>
<td>(m3)</td>
<td>equivalent cloud variable, see Gas monitor region</td>
</tr>
<tr>
<td>Q9</td>
<td>(m3)</td>
<td>equivalent cloud variable, see Gas monitor region</td>
</tr>
</tbody>
</table>
### Table 4.24: SIDS data-name identifiers of monitor region time independent parameters for monitor region volumes monitoring equivalent gas clouds

<table>
<thead>
<tr>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters/volnet</td>
<td>(m3)</td>
<td>net volume of region</td>
</tr>
<tr>
<td>Parameters/voltot</td>
<td>(m3)</td>
<td>total volume of region</td>
</tr>
<tr>
<td>Parameters/erlo</td>
<td>(-)</td>
<td>lower flammability limit defined for region</td>
</tr>
<tr>
<td>Parameters/erhi</td>
<td>(-)</td>
<td>upper flammability limit defined for region</td>
</tr>
<tr>
<td>Parameters/lfl</td>
<td>(-)</td>
<td>mixture lower flammability limit</td>
</tr>
<tr>
<td>Parameters/ufl</td>
<td>(-)</td>
<td>mixture upper flammability limit</td>
</tr>
<tr>
<td>Parameters/VEmax</td>
<td>(-)</td>
<td>maximum volume expansion ratio</td>
</tr>
<tr>
<td>Parameters/VSmax</td>
<td>(-)</td>
<td>maximum expansion ratio defined as $VS = (VE-1) \cdot ER_{fac}$</td>
</tr>
<tr>
<td>Parameters/FUmix</td>
<td>(-)</td>
<td>scaling of the fuel concentration</td>
</tr>
</tbody>
</table>

### Table 4.25: SIDS data-name identifiers of monitor region variables monitoring LFL percentiles

<table>
<thead>
<tr>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vol$&gt;$10%lfl_levels</td>
<td>(m3)</td>
<td>volume of gas above 10% LFL</td>
</tr>
<tr>
<td>Vol$&gt;$20%lfl_levels</td>
<td>(m3)</td>
<td>volume of gas above 20% LFL</td>
</tr>
<tr>
<td>Vol$&gt;$30%lfl_levels</td>
<td>(m3)</td>
<td>volume of gas above 30% LFL</td>
</tr>
<tr>
<td>Vol$&gt;$40%lfl_levels</td>
<td>(m3)</td>
<td>volume of gas above 40% LFL</td>
</tr>
<tr>
<td>Vol$&gt;$50%lfl_levels</td>
<td>(m3)</td>
<td>volume of gas above 50% LFL</td>
</tr>
<tr>
<td>Vol$&gt;$60%lfl_levels</td>
<td>(m3)</td>
<td>volume of gas above 60% LFL</td>
</tr>
<tr>
<td>Vol$&gt;$70%lfl_levels</td>
<td>(m3)</td>
<td>volume of gas above 70% LFL</td>
</tr>
<tr>
<td>Vol$&gt;$80%lfl_levels</td>
<td>(m3)</td>
<td>volume of gas above 80% LFL</td>
</tr>
<tr>
<td>Vol$&gt;$90%lfl_levels</td>
<td>(m3)</td>
<td>volume of gas above 90% LFL</td>
</tr>
<tr>
<td>Vol$&gt;$100%lfl_levels</td>
<td>(m3)</td>
<td>volume of gas above 100% LFL</td>
</tr>
<tr>
<td>Integral</td>
<td></td>
<td>unit and the meaning of the variable depends on the input, as described in Line monitor output specification</td>
</tr>
</tbody>
</table>

### Table 4.26: SIDS data-name identifiers of monitor region time independent parameters for monitor region volumes monitoring LFL percentiles

<table>
<thead>
<tr>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters/volnet</td>
<td>(m3)</td>
<td>net volume of region</td>
</tr>
<tr>
<td>Parameters/voltot</td>
<td>(m3)</td>
<td>total volume of region</td>
</tr>
<tr>
<td>Parameters/lfl_levels</td>
<td>(-)</td>
<td>reference LFL value for LFL percentiles timeseries</td>
</tr>
<tr>
<td>Parameters/integral_scale</td>
<td>(-)</td>
<td>scaling factor used to scale Integral (inverse from the denominator defined in Line monitor output specification)</td>
</tr>
</tbody>
</table>

Time series variables specific to Pool model can be accessed by the name of the variable as shown in the example below:

```python
import flacs.data
dataset = flacs.data.load_dataset("000000.cgns")
data = dataset.get_timeseries_data("PoolMeanTemperature").data)
```

SIDS data-name identifiers that can be used as the argument are listed in the table below:
4.7 Files in FLACS-CFD

Table 4.27: SIDS data-name identifiers of time series specific to Pool

<table>
<thead>
<tr>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PoolMeanTemperature</td>
<td>(K)</td>
<td>mean temperature of pool</td>
</tr>
<tr>
<td>PoolMass</td>
<td>(kg)</td>
<td>pool mass</td>
</tr>
<tr>
<td>PoolArea</td>
<td>(m2)</td>
<td>pool area</td>
</tr>
<tr>
<td>PoolEquivalentDiameter</td>
<td>(m)</td>
<td>pool equivalent diameter</td>
</tr>
<tr>
<td>PoolMassEvaporated</td>
<td>(kg)</td>
<td>total mass evaporated from pool</td>
</tr>
<tr>
<td>PoolMassRate</td>
<td>(kg/s)</td>
<td>pool mass rate</td>
</tr>
<tr>
<td>PoolRateEvaporation</td>
<td>(kg/s)</td>
<td>mass rate, evaporating gas from pool</td>
</tr>
<tr>
<td>PoolRateEvaporatedPerAreaBurgess</td>
<td>(kg/s/m2)</td>
<td>burnt rate, Burgess equation</td>
</tr>
<tr>
<td>PoolMassEvaporatedPerArea</td>
<td>(kg/m2)</td>
<td>total mass per area evaporated from pool</td>
</tr>
<tr>
<td>PoolRateEvaporationPerArea</td>
<td>(kg/s/m2)</td>
<td>mass rate per area, evaporating gas</td>
</tr>
<tr>
<td>PoolPowerPerAreaGroundConductive</td>
<td>(W/m2)</td>
<td>power per area, conduction from ground</td>
</tr>
<tr>
<td>PoolPowerPerAreaGroundConvective</td>
<td>(W/m2)</td>
<td>power per area, convection from ground</td>
</tr>
<tr>
<td>PoolPowerPerAreaGroundTotal</td>
<td>(W/m2)</td>
<td>power per area, total from ground</td>
</tr>
</tbody>
</table>

All other time series variables can be accessed by the name of the variable as shown in the example below:

```python
import flacs.data
dataset = flacs.data.load_dataset("000000.cgns")
data = dataset.get_timeseries_data("PressureMax").data)
```

SIDIS data-name identifiers that can be used as the argument are listed in the table below:

Table 4.28: SIDS data-name identifiers of other time series

<table>
<thead>
<tr>
<th>SIDS data-name identifier</th>
<th>Units</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFLC</td>
<td>(-)</td>
<td>actual sound velocity based CFL number (in Flacs2 in CFL log file)</td>
</tr>
<tr>
<td>CFLV</td>
<td>(-)</td>
<td>actual advective velocity based CFL number (in Flacs2 in CFL log file)</td>
</tr>
<tr>
<td>VelocityMax</td>
<td>(m/s)</td>
<td>maximum velocity in the computational domain</td>
</tr>
<tr>
<td>PressureMin</td>
<td>(barg)</td>
<td>the maximum pressure in the computational domain (in Flacs2 in Pressure log file)</td>
</tr>
<tr>
<td>PressureMax</td>
<td>(barg)</td>
<td>the minimum pressure in the computational domain (in Flacs2 in Pressure log file)</td>
</tr>
<tr>
<td>PressureMean&lt;-1mbarg</td>
<td>(barg)</td>
<td>the accumulated mean negative pressure for pressures less or equal to -0.001 barg in the computational domain (in Flacs2 in Pressure log file)</td>
</tr>
<tr>
<td>PressureMean&gt;+1mbarg</td>
<td>(barg)</td>
<td>the accumulated mean positive pressure for pressures larger or equal to 0.001 barg in the computational domain (in Flacs2 in Pressure log file)</td>
</tr>
</tbody>
</table>

4.7.2.2 Scalar-time output file (Flacs2 only)

Binary output file containing the value of parameters defined in the cs-file in the specified monitor points, required by Flowvis for scalar-time plots (further details can be found in section Scalar Time plot)

File name template: r1000000.dat3

For briefness this file may be called the r1-file or scalar-time output file hereafter. The values of selected
variables for scalar-time output at selected locations (monitor points) are written to the r1-file at a given iteration interval.

In the case of a simulation restart the r1-file will by default be deleted (i.e. old results are lost) when using the 2.X file format (but it is kept when using the 1.2 file format). If you want to keep the old results at restart you must set KEEP_OUTPUT=.TRUE. using a job file. During a LOAD, an existing r1-file will be scanned up to the time recorded on the specified dump file before any new results are written. It is usually wise to make a copy of the original r1-file before a load/rerun is performed.

The r1-files generally require modest amounts of disk space. The number of bytes in one data record can be calculated from the following expression (the 1.2 and 2.X file formats have approximately the same record size):

\[ \text{RecordSize} = 4 \times (2 + \text{SUM}) \]

Where SUM is the total number of references made to monitor points and panels.

4.7.2.3 Field output file (Flacs2 only)

Binary output file containing the value of parameters defined in the cs-file in all grid cells, required by Flowvis for 2D and 3D plots (further details can be found in sections 2D cut plane plot and 3D plot).

File name template: r3000000.dat3

For brevity this file may be called the r3-file or field output file hereafter. The whole matrix of each selected variable for field output is written to the r3-file, when triggered by one or more events (or signals). Simulation start and stop will trigger output as will the event of passing certain fuel levels and time values. Runtime specified events may also trigger output. A message is issued on the log file(s) for each event that triggers output. Several events may happen to trigger output simultaneously, this will generate just one instance of output.

In case of a simulation restart, the r3-file will by default be deleted (i.e. old results are lost) when using the 2.X file format (but it is kept when using the 1.2 file format). If you want to keep the old results at restart you must set KEEP_OUTPUT=.TRUE. using a job file. During a LOAD, an existing r3-file will be scanned up to the time recorded on the specified dump file before any new results are written. It is usually wise to make a copy of the original r3-file before a load/rerun is performed.

The r3-files generally require large amounts of disk space. The number of bytes in one data record can be calculated from the following expression (the 1.2 and 2.X file formats have approximately the same record size):

\[ \text{RecordSize} = 4 \times (2 + \text{NX} \times \text{NY} \times \text{NZ}) \times \text{NVAR} \]

NVAR number of variables specified
NX, NY, NZ matrix dimensions

4.7.2.4 Simulation log file - rt-file

File name template: rt000000.dat3

For brevity this file may be called the rt-file or log file hereafter. It contains a log of the messages issued during a simulation. It is wise to check if any error messages have been issued. Error messages are identified by "***" at the beginning of the message, plain messages are introduced by a space ( ) or a number sign (#).

In the case of a simulation restart the rt-file will be deleted. It is usually wise to make a copy of the original rt-file before a load/rerun is performed.

The information on the log file is also written to a connected terminal (standard output) during the simulation.

4.7.2.4.1 Volume of calculation domain blocked by geometry

Information about the volume of the calculation domain blocked by the geometry is included in the rt-file. The following text (example for a given geometry and simulation) is given in the rt-file:
4.7 Files in FLACS-CFD

Volume of calculation domain:
total = 3000.000 m³
blocked = 192.997 m³

In this example it says that the total volume of the computational domain is 3000.000 m³, and a part of this volume, 192.997 m³, is blocked due to the geometry. The rest of the volume (3000.000 m³ - 192.997 m³) is open for fluid.

4.7.2.4.2 Flammability limits  Information about the flammability limits is included in the rt-file.
Among others the following text (example for a given fuel composition) is given in the rt-file:

Flammability limits, equivalence ratios (ER):
LFL %LFL 0.530 100.0
TOP %LFL 1.152 217.3
UFL %LFL 2.200 415.1

The values of the equivalence ratio corresponding to LFL and UFL are given. The equivalence ratio, ER [-], is well defined for a flammable fuel-air mixture. The equivalence ratio ER is defined as the ratio of mass of fuel to mass of oxygen (for the fuel-air mixture considered), divided by the ratio of mass of fuel to mass of oxygen at the stoichiometric concentration. Thus when \( ER = 1.0 \), the concentration of fuel is stoichiometric, and when \( ER = 0.0 \) the mass of fuel is zero.

The log file in this example (for the fuel-air mixture considered) tells that, according to the modelling in FLACS-CFD, the Lower Flammability Limit (LFL) is for \( ER_{LFL} = 0.530 \), the maximum laminar burning velocity is for \( ER_{TOP} = 1.152 \) (the value of \( ER_{TOP} \) is in general close to the stoichiometric condition \( ER=1.0 \)), and the Upper Flammability Limit (UFL) is for \( ER_{UFL} = 2.200 \). The value of

\[
100 \cdot \left( \frac{ER_{UFL}}{ER_{LFL}} \right) = 100 \cdot \left( \frac{2.200}{0.530} \right)
\]

is in this example calculated to 415.1 which is also stated in the log file (value of ER at UFL expressed as "%LFL"; the ratio of \( ER_{UFL} \) to \( ER_{LFL} \) in percent). Similarly the value of

\[
100 \cdot \left( \frac{ER_{TOP}}{ER_{LFL}} \right) = 100 \cdot \left( \frac{1.152}{0.530} \right)
\]

is in this example calculated to 217.3 which is also stated in the log file.

4.7.2.4.3 Amount of fuel inside computational domain  The log file will in general contain information for each time step similar to (example for job no. 654321 including the first two time steps; ITER=0,1):

```
ITER TIME P_MAX V_MAX IP FUEL RATE DT PMEAN
[##] [s] [barg] [m/s] [##] [kg] [kg/s] [s] [barg]
0 0.000000 0.000 2.4 0 0.00E+00 0.00E+00 1.13E-01 0.000
# OPENED 020000 :./r1654321.dat3
# OPENED 030000 :./r3654321.dat3
# writing ...
1 0.056643 0.007 2.6 3 1.97E+00 1.74E+01 1.13E-01 0.001
```

The amount of "FUEL [kg]" is the total amount of gaseous fuel composition inside the computational domain for the time step considered.

Note:

If inert gases are included in the fuel composition, then these are included in the total amount. Thus, if for example carbon dioxide (CO2) is included in the fuel gas composition, then CO2 is also included in the reported amount of "FUEL [kg]" in the log file rt654321.dat3.
The "RATE [kg/s]" reported for each time step in the log file concerns the change of total amount of fuel composition per unit time inside the computational domain. For dispersion scenarios it may for example happen that one has a scenario where a jet leak with steady mass rate of fuel is balanced by the same amount of fuel per unit time leaving the computational domain (through an open boundary of the computational domain). For such a scenario with steady-state, the reported "RATE [kg/s]" will be zero, corresponding to no change in the total amount of fuel composition inside the computational domain. In explosion scenarios one will typically see reported time steps where the "RATE [kg/s]" is negative due to fuel being consumed by combustion.

4.7.2.5 Simulation log file - tt-file

Output log containing the same output as the rt-file, with the addition of output messages from the operating system. It can be useful to monitor the writing of this file during simulations:

Linux:
> tail --f tt010100

Remarks:
This file is not created on Windows.

4.7.2.6 Other simulation log files (Flacs2 only)

File name template: rt000000.dat3.*

Several additional log files will be generated when using Flacs2. Most of these files are textual and can be plotted in common tools such as gnuplot on Linux or Excel on Windows. The contents of each file is explained at the start of the file, below you will only find a brief explanation of the new log files:

<table>
<thead>
<tr>
<th>File name</th>
<th>Contents of file</th>
</tr>
</thead>
<tbody>
<tr>
<td>rt000000.CFL</td>
<td>CFL-numbers and time step log file</td>
</tr>
<tr>
<td>rt000000.FUEL</td>
<td>Amount of fuel log file</td>
</tr>
<tr>
<td>rt000000.MASS</td>
<td>Total mass log file</td>
</tr>
<tr>
<td>rt000000.P</td>
<td>Pressure log file</td>
</tr>
<tr>
<td>rt000000.dat3</td>
<td>The usual simulation log file</td>
</tr>
</tbody>
</table>

The most relevant file is the "rt000000.FUEL" file, where the amount of fuel in the simulation volume (total amount and combustible amount) is logged.

4.7.2.6.1 CFL log file

Output log of CFL numbers:

```
# Effective CFL numbers:
# sound convection diffusion
# TIME(s)   CFL_C(-)   CFL_V(-)   CFL_D(-)
```

The columns in the CFL log file explained:

- "TIME (s)" is the intermediate time between current time and previous time
- "CFL_C (-)" is the Courant-Friedrichs-Levy number based on sound velocity. For more details, cf. CFLC.
- "CFL_V (-)" is the Courant-Friedrichs-Levy number based on fluid flow velocity. For more details, cf. CFLV.
- "CFL_D (-)" is the Courant-Friedrichs-Levy number based on diffusion velocity. For more details, cf. Time step for the incompressible solver.
4.7 Files in FLACS-CFD

4.7.2.6.2 Fuel log file
A log of the amount of fuel in the gas monitor region (i.e. the whole or a subset of the simulation volume) and several derived measures is stored in the fuel log file rt<jobno>.FUEL. Note that there is a difference between the parameter FUEL [kg] reported in the general simulation log file rt<jobno>.dat3 and the parameter TOT_FUEL [kg] reported in the fuel log file rt<jobno>.FUEL. For an explosion simulation, the reported FUEL [kg] in the general log file rt<jobno>.dat3 includes only unburned, gaseous fuel (i.e. FUEL [kg] is distinguished from combustion products). During an explosion simulation, FUEL will in general decrease to almost zero when the simulation stops based on the automatic criterion (assuming that no additional fuel is added during the explosion). TOT_FUEL, on the other hand, will remain the same, unless unburned or burned fuel leaves the gas monitor region. This means that for example for a stoichiometric propane-air cloud that is ignited inside a totally closed cube, FUEL will decrease during the simulation, whereas TOT_FUEL as a function of time for a standard gas monitor region (covering the same volume as the complete computational domain, i.e. the closed cube) remains the same during the whole simulation, corresponding to the initial value of FUEL.

The value of TOT_FUEL for the gas monitor regions is based on the transport equation for FMIX without taking into account how much fuel is burnt during the simulation. The mixture fraction FMIX measures how much the gas composition corresponding to ER0 has mixed with the gas composition corresponding to ER9 during the simulation. For the example above, the value of FMIX is identically 1.0 during the whole simulation: the stoichiometric propane-air mixture (ER0 = 1.0) fills the whole cube and does not mix at all (neither initially or during the combustion process) with any ambient air (ER9 = 0.0); combustion alone does not imply mixing between the gas compositions corresponding to ER0 and ER9.

In summary, for pure dispersion simulations, FUEL and TOT_FUEL are the same, but for explosion simulations they normally differ.

The fuel log file also contains the values of FlamMass, FlamVol, GasMass, ERfacMass, and the quantities Q1, ..., Q6,Q7,Q8,Q9.

When you have defined custom gas monitor regions, there will be one additional fuel log file per region, named rt<jobno>.MON-Monitor_region, with similar information for the respective region as found in the standard fuel log file rt<jobno>.FUEL.

4.7.2.6.3 Mass log file
Output log of the mass in the simulation volume:

# TOTAL INITIAL MASS = 1.1835E+03
# time(s) level(--) added(kg) time(s) rate(kg/s) leak(kg/s) rate(kg/s) flow(kg/s) sum(kg)

The columns in the mass log file explained:

- "time (s)" is the intermediate time between current time and previous time (global time)
- "level(--)" is total mass in the computational domain (current time) divided by total initial mass in the computational domain
- "added (kg)" is total mass in the computational domain (current time) minus total initial mass in the computational domain
- "time (s)" is the intermediate time between current time and previous time (from leaks)
- "rate (kg/s)" is total mass in the computational domain (current time) minus total mass in the computational domain (previous time) divided by the timestep length
- "leak (kg/s)" is the accumulated leak rate for all leak faces at current timestep
- "rate (kg/s)" is the accumulated leak rate for all leak points at current timestep
- "flow (kg/s)" is the accumulated leak rate for all leak directions at current timestep
- "sum (kg)" is the accumulated added mass from all leaks, over the simulation time
4.7.2.6.4 Pressure log file  Output log of pressure:

```
# Pressure statistics:
# minimum maximum mean<0.001 mean>+0.001
# TIME(s) P(barg) P(barg) P(barg) P(barg)
```

The columns in the pressure log file explained:

- "time (s)" is the intermediate time between current time and previous time, global time information
- "minimum P (barg)" is the minimum pressure in the computational domain
- "maximum P (barg)" is the maximum pressure in the computational domain
- "mean < -0.001 P(barg)" is the accumulated mean negative pressure for pressures less or equal to -0.001 barg in the computational domain
- "mean > +0.001 P(barg)" is the accumulated mean positive pressure for pressures larger or equal to 0.001 barg in the computational domain

4.7.2.7 Simulation dump file (Flacs2 only)

Dump files contain a 'snapshot' of the simulation at a selected time defined in the cc-file. File name template: `rd000000.n000`. The index number in the file name extension identifies the dump. For briefness this file may be called the rd-file or dump file hereafter. It contains the information necessary to restart a simulation from a given time required that the initial input data have not been changed drastically. A typical use of the dump file is to start an explosion simulation using the resulting gas cloud from a dispersion simulation.

**Note:**

There may be more than one rd-file.

**Attention:**

If you want to append new data to existing result files, do not change the output specification or the grid or the geometry when using a dump file! It is possible to convert the dump file data to restart simulation on a smaller grid by using the rdfile utility.

4.7.2.8 Monitor file (Flacs2 only)

Based on the cs.MON file, Flacs can write one or several additional output files. As an example, the cs.MON section uses an input file with the following content:

```
VERSION 1.0
# This is a comment
volume(name="123455",start=138,-21,32,end=198,21,48,output="FUEL(see=1.0,mix=0.8)")
volume(name="123456",start=138,-21,32,end=198,21,48,output="FUEL(see=1.0,mix=1)")
volume(name="123457",start=138,-21,32,end=198,21,48,output="FUEL(see=1.0,mix=1.2)")
```

Faced with this input, Flacs will write three output files (text files) `rt010100.MON.123455`, `rt010100.MON.123456` and `rt010100.MON.123457` based on different fuel concentrations as explained in the section on the frozen cloud concept. An additional example follows in the subsection below.

**See also:**

- cs.MON file section.
Note:
The reported TOT_FUEL [kg] in the rt<jobno>.MON.Monitor_region differs from the reported FUEL [kg] in the rt<jobno>.dat3 simulation log file, see section Fuel log file.

Note:
The r3file utility also has an option to extract line monitor output from the r3-file (rather than an rt.MON file), assuming the r3-file contains the necessary data (cf. additional guidance related to the r3file utility). Example:

```
run r3file r3000000.dat3 'line(name="A3",start=0,6,1,end=40,6,1, output="FUEL(ER)")' force
```

4.7.2.8.1 Example: volume with fuel concentration above a defined level

The volume [m$^3$] inside a region (the whole computational domain or a subregion) where the fuel concentration is above a defined level, can be output using a monitor file by giving a cs.MON file (here cs010100.MON) with the following content:

```
VERSION 1.0
volume(name="LFL1",start=3300,3800,44,end=3550,4050,94, output="FUEL(LFL,ER)"
volume(name="LFL2",start=3300,3800,44,end=3550,4050,94, output="FUEL(LFL=1.0,ERUFL=1E+30)"
volume(name="LFL3",start=3300,3800,44,end=3550,4050,94, output="FUEL(MoleLFL=0.5,MoleUFL=1.0)"
volume(name="LFL4",start=3300,3800,44,end=3550,4050,94, output="FUEL(MassLFL=0.4,MassUFL=1.0)"
```

Note that the input lines are split for readability here in the manual, every volume entry must appear on a single line.

In this example four extra output files are made by flacs: rt010100.MON.LFL1, rt010100.MON.LFL2, rt010100.MON.LFL3, and rt010100.MON.LFL4.

The first few lines from the output file rt010100.MON.LFL1 are in this example given by:

```
# LFL = 0.3886
# TIME 0.1000 0.2000 ... 0.9000 1.0000
0.0000E+00 0.0000E+00 0.0000E+00 ... 0.0000E+00 0.0000E+00
1.4391E-02 0.0000E+00 0.0000E+00 ... 0.0000E+00 0.0000E+00
4.3173E-02 0.0000E+00 0.0000E+00 ... 0.0000E+00 0.0000E+00
```

The first column contains the simulated time for each output.
The second column shows the volume [m$^3$] inside the region specified in the cs.MON file where the fuel concentration is equal or higher than 0.1 times the equivalence ratio (ER) corresponding to LFL (Lower Flammability Limit). In this example ER_LFL = 0.3886, as specified in the first line.
The following columns specify the volume for higher percentages of ER_LFL: column 3 for 20% times ER_LFL, and so on. In the last column (1.0=100%), the volume inside the specified region with fuel concentration equal or higher than the value of the ER corresponding to LFL, is specified.

For output file rt010100.MON.LFL2 the difference compared to rt010100.MON.LFL1 is that the output is based on ER_LFL = 1.0 and not the default value of ER_LFL used in the combustion model in Flacs.

For the output file rt010100.MON.LFL3 the concentration is measured by the molar fraction of the fuel composition rather than the equivalence ratio, triggered by output="FUEL(MoleLFL=0.5,MoleUFL=1.0)".

For the output file rt010100.MON.LFL4 the concentration is measured by the mass fraction of the fuel composition rather than the equivalence ratio or the molar fraction as indicated by output="FUEL(MassLFL=0.4,MassUFL=1.0)".
Warning:

Always specify the concentrations at the lower flammability limit (LFL) lower than the upper one (UFL) in the monitor file (when default LFL and UFL are used this is automatically the case).

Note:

The setting of the flammability limits for a monitor volume is only used for the computation of the output for the relevant rt.MON file; it does not influence the flammability limits in the combustion model in Flacs.

4.7.2.9 Simulation save file

Temporary file generated by FLACS-CFD in some situations. For briefness this file may be called the rx-file or save file hereafter. It is a temporary file that FLACS-CFD uses to store intermediate data for output. There may be more than one rx-file. These files are needed when time integrals of whole matrices are to be calculated. It avoids using large amounts of core memory but accessing large files is slow compared to memory access. The rx-files are only written if you have specified field output of variables that need temporary file storage. These files are deleted when the simulation terminates normally. If the simulation does not finish correctly these files will exist after program stop and they must then be deleted manually. Existing temporary files may confuse the program at start-up. Below is a list of the variables that require temporary file storage when specified for field output:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP_IMP</td>
<td>Pressure impulse</td>
</tr>
<tr>
<td>NUDRAG_IMP</td>
<td>Drag-impulse component x-direction</td>
</tr>
<tr>
<td>NVDRAG_IMP</td>
<td>Drag-impulse component y-direction</td>
</tr>
<tr>
<td>NWDRAG_IMP</td>
<td>Drag-impulse component z-direction</td>
</tr>
<tr>
<td>NDRAG_IMP</td>
<td>Drag-impulse value</td>
</tr>
<tr>
<td>NRESID</td>
<td>Mass residual in continuity equation</td>
</tr>
</tbody>
</table>

It is advised that the above listed variables are not specified for field output. The file handling will increase the execution time for the program drastically!

4.8 Warnings and error messages

Flacs will write any error or warning messages to the Simulation log file - tt-file. The following sections give a description of the most common warnings and errors, and possible causes and workarounds.

4.8.1 Mass residual

The message ”*** MASS RESIDUAL =...” indicates that there are problems to achieve a converged solution. If the mass residual is not too large (should be smaller than order of 1/10) or it does not last for many iterations it may be ignored. The built-in limit for the mass residual is rather strict to enable an accurate solution over long time, a temporary violation of the limit does not necessarily lead to a wrong result. Mass residuals may happen for a range of reasons, but occur most often in dispersion calculations. For explosion calculations mass residuals are mainly seen when there are very high pressures in a simulation and the flame is going through partly porous areas. This can sometimes be solved by a setup file changing the values of ZERO_APOR and ZERO_VPOR (details).

For a combined dispersion and explosion simulation where the explosion simulation is started from a dump file, you can get a mass residual error message if you change the equivalence ratio between the two simulations. For such simulations, when a non-homogeneous fuel-air cloud calculated in a dispersion...
4.8 Warnings and error messages

Simulation is ignited in the explosion simulation, it is advised to keep the same value of ER0 for both the dispersion simulation and the subsequent explosion simulation. If you need to change the equivalence ratio, for example from ER0=1E+30 (pure fuel release) in the dispersion simulation to ER0=1 for the subsequent explosion simulation, you should use the convert option of the rdfile utility program to ensure that the output enthalpy field is converted consistently for the changed value of ER0.

For dispersion calculations there are a number of reasons, and possible solutions, for mass residuals. Starting from version 22.2, the simulator automatically rewinds and restarts the simulation with a smaller time step if a large mass residual condition occurs. The restart point is selected among the periodic checkpoints that are saved to file during the execution of the simulation. If no suitable checkpoint is found, the simulation is restarted from the initial state. After a rewind-restart, when the simulation progress to the time when the large mass residual condition was detected, the time step is gradually increased towards the original value determined by CFL numbers and other time step constraints. Multiple automatic rewind-restarts may be performed during a simulation if the mass residual condition occurs more than once. Each time a rewind-restart is performed, the time step is reduced by a constant factor, thus, the time step can gradually decrease after subsequent rewind-restart events if these happen before the time step recovered to the original value. If the reduction (ratio between the uncorrected time step and the time step applied after rewind-restart) is more than a factor 8, the simulation is stopped with error: '# STOPPED DUE TO LARGE MASS RESIDUAL'. In such cases, when the automatic rewind-restart does not solve the large mass residual condition, you should check the simulation setup for problems or specific conditions, according to the following guidelines, and apply the suggested corrections.

Before you try the solutions, you should try to identify where in the simulation domain the mass residual happens (look at the coordinates of its location, and thereafter check in Flowvis (note that Flowvis may be +/- one grid cell shifted during visualisation compared to Flacs).

- Very early crash when wind/ventilation is started:
  - Hard to adjust wind to geometry, solution that often works is to define low values for initial turbulence (CHAR VEL=wind speed, RELATIVE TURB = 0.01 and LENGTH SCALE = 0.01). This may help the simulation though the transient start-up, better turbulence parameters will later come from the boundary.

- Crash when leak starts (or stops):
  - May be a too transient leak start (solution is to modify profile manually and make a longer "ramp") or something may be wrong in the leak definition (e.g. too high flow relative to grid cell size or significant porosities in leak volume so pressure is building up).

- Sudden crashes:
  - Stability class A, B, C may give problems, avoid using them

- Too long turbulence length scale relative to grid cell size:
  - Try using Pasquill class F instead of D and E.
  - If still a problem, do not use a Pasquill class, but define turbulence parameters in the wind boundary condition, with intensity 0.05 to 0.10 and length scale not more than the size of the smallest grid cell (you may try to exclude refined grid cells).

- Too thin grid cells:
  - Very narrow grid cells (e.g. 10cm in the refined zone when the normal grid is 1m and stretched to the boundaries to 10m) often give problems. Try first to reduce the stretching towards the boundaries (e.g. from 10m to 3m) and thereafter, if necessary, to increase the 10cm minimal size. However, such cases are not always easy to solve.

- In cases with stretched grids, reduce the stretching and make sure FLUX_CONTROL=2 (details)

- Pool model:
– Pool models may lead to some additional stability issues.

• Porosities:
  – Dispersion calculations in partly porous regions may also sometimes give problems, and it may help to increase the limit for small porosities as described for explosions above.

• When the simulation is started from a dump file created using the rdfile utility, and the simulation run demands a small initial time step for the numerical solver to be stable:
  – You can explicitly set the delta time for the time step being stored in the out_file dump file, by using the option dt=value for the rdfile utility, cf. the section on rdfile details.

• Boundary conditions may also lead to stability problems in some situations. If you suspect this to be the case, try to change passive outflow (NOZZLE) at parallel boundaries to WIND and move the boundaries further away.

4.8.2 Leak excess area

FLACS-CFD prints the warning: "*** LEAK EXCESS AREA ..." if the expanded leak area for a point leak is greater than the cell face area for the outflow. This can occur even if the expanded leak area defined in the scenario inputs is smaller than the area of a grid cell, as described below.

The release of a high-density gas, a cold jet release, and releases within partially porous volumes may trigger this warning. At the beginning of a jet release, the density of the released fuel is interpolated between the effective release point (the centre of the release cell, which is the grid cell where the release is located) and the first downstream grid cell. For a release of a heavy pure fuel, this can lead to a density at the face of the release cell that is initially lower than the density of the pure fuel. To achieve a fixed release rate with a fixed flow velocity, FLACS-CFD then increases the effective leak area, however it is only possible to increase the effective area up to the area of the release cell face. If the required increase is greater than this, then the warning "*** LEAK EXCESS AREA ..." is triggered.

The solution for this warning is to:

• increase the grid cell size where the leak is defined

• or redefine the leak so that the leak area is decreased

4.8.2.1 Warnings and error messages for area leaks

When the size of the leak box cannot accommodate the prescribed leak shape, FLACS-CFD will issue a warning or an error, depending on the discrepancy between the desired area and the maximum available one. If the difference is small, then Flacs will give a warning like:

*** WARNING: leak number % got too large area (compensated), factor=1.001

and the computation will continue with the flux density over the leak area increased to match the total flux prescribed. This may be caused by the leak box snapping to the nearest grid line making the leak box smaller than extended. Adjusting the boundaries of the leak box to be in grid planes and possibly extending the leak box by one cell may help to overcome this problem and avoid the warning.

If there is a large discrepancy between the requested and possible leak area then the simulator will stop and print an error like:

*** ERROR: leak number % got too large area, factor=1.234
4.9 Potential bugs and problems with Flacs

A possible cause is that the area passed in the leak file has been used to dimension a rectangular leak box while the leak is supposed to be elliptic, so that it cannot occupy the whole leak box. In this case, changing the shape or increasing the size of the leak box may help.

See also:

The leak box parameters are described in the area leaks section.

If a leak box is specified but neither the leak file nor the TYPE definition in the scenario file contain an :AreaLeak(...) clause then Flacs will give an error message like

*** ERROR: missing ":AreaLeak(...)"

4.8.3 CV too small

If there are overlapping grid lines, an error like the following will be displayed:

ERROR: xcv(2) is too small
ERROR: xcv(291) is too small

In this case there are overlapping grid lines in the X direction. To investigate and fix the problem check Grid →List in CASD, and look for grid lines with the same X position. Delete one of them, save and rerun the porosity calculation. In case the issue is present along the Y or Z axis, the error message concerns ycv or zcv, respectively.

4.8.4 Errors reading the cs-file

If there is an error in the cs-file, the error message: "Error: Reading file cs<jobno>.dat3 file", will be issued. This error message is often caused by formatting or syntax errors in the scenario file; the following approaches may help to correct the cs-file:

- Open the rt-file or the tt-file in a text editor. These files should show a more complete error message, and possibly state the line where the error occurs.
- Sometimes one can isolate a syntax error by opening the scenario file in CASD, saving it to a different job number and comparing the two cs-files.

4.8.5 Mixing height <= 0 m

When using a logarithmic wind profile and Pasquill class D-F the wind velocity must be greater than zero, otherwise you get the error message *** ERROR: Mixing height <= 0 m. In other words, a Pasquill class can be combined with WIND boundary conditions only when the velocity on the WIND boundaries is set to a non-zero value. If your dispersion/explosion scenario has a WIND boundary condition with zero velocity, then it may suffice to set a positive wind velocity for the boundary condition. Alternatively, to avoid this error, change the WIND boundaries to NOZZLE and set the Pasquill class to "NONE".

A related error message for Pasquill class A-C is *** ERROR: Surface heat z1 <= 0, if the surface heat flux is not given correctly.

4.9 Potential bugs and problems with Flacs

This chapter contains a list of potential bugs or problems associated with the CFD simulator Flacs, and some possible workarounds.

See also:

Modelling and application limitations are discussed in a later section.
4.9.1 Significant overprediction of lower flammability limits for hydrogen+inert mixtures

In previous versions of Flacs there was an issue with hydrogen+inert (Nitrogen, CO\(_2\) etc.) mixtures. Flacs would significantly overestimate the lower flammability limit (LFL) for the mixture in air. For higher inert contents, this overprediction may be a factor 2 or higher. When using these mixtures in dispersion simulations the flammable cloud volume would be significantly underpredicted, which was non-conservative. It was caused by the applied gas mixture scaling algorithm, which is not accurate enough for hydrogen inert mixtures. The issue is present in all previous versions of FLACS-CFD (v10.8r2 and earlier). It is important to mention that pure hydrogen (more widely used) and hydrocarbon+inert mixtures were both not affected by this issue. This issue was resolved in FLACS-CFD v10.9, by incorporating a lookup table for hydrogen+nitrogen mixtures. This approach was verified against experimental laminar burning velocity data (Konnov, 2018) and validated against a two series of small scale explosion experiments (Mogeleg and small scale 3D Corner (with both pure H\(_2\) and 75 vol\% H\(_2\) + 25 vol\% N\(_2\) mixtures)). Current limitations for hydrogen-inert mixtures are the following:

- Only pure hydrogen-nitrogen mixtures (in air) are supported. Other hydrogen mixtures with more flammable components (e.g. hydrogen+nitrogen+methane) or other inert components (e.g. CO\(_2\)) are not supported and Flacs will use the original mixing algorithm and may give incorrect LFL and/or UFL values.
- The model should be accurate for hydrogen-nitrogen fuel mixtures with up to 90 vol\% nitrogen.

If any previous studies where run using hydrogen+inert mixtures and a FLACS-CFD version v10.8r2 or before, we strongly recommend to review the LFL values calculated by Flacs in the simulation log files and assess impact in case it differs from the expected LFL value. For further guidance and information please contact FLACS-CFD support.

4.9.2 Underprediction of lift-off length for jet fires

Flacs may underpredict flame lift off length for vertical jet fires. In our validation cases and in most real-life cases this will be conservative, as the flame will start at a lower height and the heat radiation will be higher at ground level. However, in cases where the point of interest is at or above flame height, this may result in an underprediction results in those points. This issue is present in all versions of FLACS-Fire. In FLACS-CFD v10.8 an experimental model using the chemical time scale was included to improve lift off length calculations. However, this model has not yet been validated and is therefore off by default. For additional guidance and information please contact FLACS-CFD support.

4.9.3 Heavy hydrocarbons C\(_5\)H\(_{12}\) and upwards

Several vapours from hydrocarbons that are heavier than butane can be modelled, but combustion properties have been copied from butane. Enthalpies for Dodecane have been copied from Decane, and this may lead to strange results if burning Dodecane. The heavy hydrocarbons have a high boiling point, so in reality they will appear more as a mist than a gas at normal atmospheric conditions, which will affect both the burning velocity and the flammability range.

4.9.4 FLACS-CFD underpredicts in empty enclosures

Flacs may underpredict overpressures resulting from explosions in “empty” enclosures (with vents) due to missing models for flame instabilities. This problem becomes less relevant as the obstacle density increases, but is under investigation by Gexcon R&D. Improved models are expected in future releases.
Chapter 5

Flowvis

Flowvis, the postprocessor for the CFD-code FLACS-CFD, is a program for visualising results from simulations of gas explosions, gas dispersion and multiphase flow.

5.1 System requirements

Please see the section on Hardware and software requirements regarding the graphics card requirements for running Flowvis.

5.2 The Flowvis main window

The current section gives a brief introduction to the main window of Flowvis. The main window is displayed all the time while Flowvis is running. The main window contains a menu bar, a tool bar, a properties sidebar, the plot area, and a time slider. If a presentation comprises several pages then these are accessed via tabs on the right of the plot area.

Figure 5.1: Flowvis main window
5.2.1 The menu bar

The menu bar contains the available menus.

1. The File menu contains commands for opening, saving, and exporting presentations.
2. The Edit menu involves commands that manipulate pages and plots.
3. The Page menu collects commands to administer pages.
4. The Plot menu offers commands for creating and editing plots.
5. The Tools menu comprises the porosity verification functionality of Flowvis.
6. The Options menu contains toggle buttons for various options and gives access to advanced settings.
7. The Help menu gives access to help and documentation about Flowvis.

5.2.2 The tool bar

The Flowvis tool bar is shown below.

![Flowvis tool bar](image)

**Figure 5.2: The Flowvis tool bar**

<table>
<thead>
<tr>
<th>Button</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>Creates a new presentation.</td>
</tr>
<tr>
<td>(b)</td>
<td>Opens an existing presentation.</td>
</tr>
<tr>
<td>(c)</td>
<td>Saves the presentation.</td>
</tr>
<tr>
<td>(d)</td>
<td>Deletes the selected plot, but puts it on the clipboard.</td>
</tr>
<tr>
<td>(e)</td>
<td>Copies the selected plot onto the clipboard.</td>
</tr>
<tr>
<td>(f)</td>
<td>Pastes a plot from the clipboard.</td>
</tr>
<tr>
<td>(g)</td>
<td>Creates a new page in the presentation.</td>
</tr>
<tr>
<td>(h)</td>
<td>Deletes the current page.</td>
</tr>
<tr>
<td>(i)</td>
<td>Sets the page layout: Several plots can be arranged in a grid on one page. Each plot can also span multiple cells of the grid.</td>
</tr>
<tr>
<td>(j)</td>
<td>Reloads the data from disk. Useful if a plot shows results from a running simulation.</td>
</tr>
<tr>
<td>(k)</td>
<td>Resets the current plot to the default view.</td>
</tr>
<tr>
<td>(l)</td>
<td>Returns to the previous view for the current plot (possible for zoom and pan operations in 1D and 2D plots).</td>
</tr>
<tr>
<td>(m)</td>
<td>Advances to the next view for the current plot (possible for zoom and pan operations in 1D and 2D plots).</td>
</tr>
<tr>
<td>(n)</td>
<td>Allows to move the plot region.</td>
</tr>
<tr>
<td>(o)</td>
<td>Enables to zoom in to the subsequently marked portion of the plot.</td>
</tr>
<tr>
<td>(p)</td>
<td>Opens the figure options dialogue.</td>
</tr>
<tr>
<td>(q)</td>
<td>Opens the Aggregation dialogue.</td>
</tr>
</tbody>
</table>

Right-clicking an empty area in the tool bar brings up a menu to hide or show the different tool bar sections, as well as the properties sidebar.  

FLACS-CFD v22.2 User’s Manual
5.2 The Flowvis main window

5.2.3 The properties sidebar

The properties sidebar shows the settings for the currently selected plot. The sidebar can be detached from the main window, or moved to a different location, by dragging it. Refer to the various plot types to see what properties can be set for each plot.

Note:

If you have closed the plot properties sidebar then you can get it back by right-clicking in the grey area of menu bar or tool bar. This will open the context menu where you can re-enable the plot properties entry.

![Plot Properties Sidebar](image)

Figure 5.3: The plot properties sidebar can be re-activated in the context menu available by right-clicking in the menu bar or tool bar.

5.2.4 The plot area

The plot area can have multiple pages, and each page can contain multiple plots. Right-clicking in an empty space in the plot area shows the menu to create a new plot. A right-click on a plot shows the context menu for that plot. By clicking "Set span" in the context menu, you can set how many cells in the plot layout the current plot will cover.

![Plot Area with Multiple Pages](image)

Figure 5.4: A presentation with four pages and a page layout with three plots.
Optionally, the top part of the plot area can be used to display a header for the current page. Using the headers can be switched on or off in the Options menu. The header will be visible on exported graphics.

![Figure 5.5: The page header with information about the plots’ origin and creation time.](image)

The content of the page header can be changed per page or for all pages in the Page Settings dialogue box from the Page menu. Choose the Header page and fill in the desired text. In particular when assigning a text to all pages, the use of variables like the page number, the name of the presentation file, or the creation time can be useful. Flowvis will replace these variables by their value in the header on each page.

![Figure 5.6: The Page Settings dialogue box allows to change the content of the page header.](image)

### 5.2.5 The time slider

The time slider has two modes that can be toggled by activating or deactivating Options → Synchronize Time. In synchronised mode (default), the time for all plots on a page is the same; the tick marks of the time slider are at the beginning values of the time steps, and clicking in the time slider selects a point in simulated time. When Synchronize Time is deactivated, the time slider selects time steps and the tick marks along the slider are equally spaced rather than according to the time step length. On a page with plots of different scenarios, the plots may then show different times, since a certain time step may refer to different times in the different datasets.

![Figure 5.7: The time slider in synchronised mode (upper) and time step mode (lower). On the right, the drop-down menu that appears when holding down the play button is shown; here the animation speed can be reduced or increased.](image)

An additional difference between the two modes of the time slider is that when playing an animation of the simulated time steps, Flowvis will aim to show the animation in real-time when Synchronize Time is
on. This means that simulations that cover a short time span (like explosions) will consist of only very few frames, depending on the speed of the computer used for visualisation. To play the animation at reduced speed (enabling the graphics card to produce more frames), left-click and hold the mouse pointer on the play button. In the drop-down menu that appears, choose one of the speed-up factors smaller than one.

### 5.2.6 Page selection via tabs

If several plots are gathered in one presentation then these are accessed via tabs along the right hand side border of the plot area. The page names can be changed either per page or for all pages by specifying a template in the Page Settings dialogue box from the Page menu. The naming template may use variable text elements like the page number, the presentation file name, or the current date. Pages can also be dragged and dropped in the tab bar to rearrange their order.

![Page Settings Dialogue Box]

Figure 5.8: The Page Settings dialogue box allows to change the names of pages as they occur on the tabs for page selection.

### 5.3 Plot types

#### 5.3.1 Scalar Time plot

A Scalar Time plot is a curve plot where the value of one or several scalar variables is plotted over a time interval. Output from several simulations and for several monitor points can be combined in one plot.
Setting up a Scalar Time plot is quite straightforward, by selecting jobs, variables, and monitor points or panels for the selected plot. Selecting jobs and variables can also be done via the context menu entry Data Selection. Other formatting options can be accessed by right-clicking into the plot or via the Edit menu. The axis ranges can be set in the Figure options dialogue.

In Flowvis, also 3D variables can be shown in Scalar Time plots. When selecting a 3D variable, the plot will contain a curve representing the median value of that variable, and a shaded area of the same colour that shows the whole range of values present at the corresponding time. The median and range are determined for the active region of the domain; the active region is set with the domain sliders in the properties sidebar. A similar range will be shown when using a 1D variable without selecting specific monitor points. In this case the minimum, maximum and median are determined based on the values from all monitor points.

5.3.2 Scalar Line plot

A scalar line plot is a 2D cartesian plot of a scalar variable, sampled along a straight line running through the simulation domain, for a specific time step. The line has to be parallel to one of the grid axes, but can be
positioned freely in the other two dimensions. In the properties sidebar for the Scalar Line plot, the scalar variable to plot can be selected in the Data Selection list. The placement and orientation of the line to sample values along, is specified by choosing which axis the line shall follow (X, Y or Z), setting the start and end position along this axis, and setting the position of the line in the other two axes.

The illustration below shows a Flowvis page with two sub-plots: a Scalar Line plot at the top, and a 2D Cut Plane at the bottom. A green dotted line drawn on top of the 2D Cut Plane illustrates the position of the sample line specified in the Scalar Line plot. The two plots are aligned in the horizontal direction (along the X-axis), as indicated by the four vertical red dotted lines drawn on top of both plots. The alignment of the two plots along the X-axis shows how the variations in scalar variable value (PROD_3D) along the line, matches in the two plots. Note that the green and red dotted lines in this figure is added on top of the Flowvis image using a graphics editor.

![Figure 5.11: Scalar Line plot on top, showing data along the green dotted line in the 2D Cut Plane plot below.](image)

### 5.3.3 2D cut plane plot

The 2D cut plane plot type shows a two-dimensional cross-section of the domain or a part of the domain, parallel to a grid plane. The plot can include output for none, one or two variables. If two variables are combined, one must be a scalar variable and the other one must be a vector variable. Scalar variables are shown using filled contours while vector variables are displayed with arrows. The plot can include output from several simulations. The grid, geometry and porosities can also be included.
Figure 5.12: 2D cut plane plot

The plot has one legend for each variable. The legends are by default positioned to the right of the plot, but may be moved around freely by dragging them with the mouse. The colour map for each variable can be customised using the Appearance window in the properties sidebar.

The legend for vector variables includes arrow lengths. The arrow length is mapped to the vector magnitude in the 2D plane, while the colour is mapped to the magnitude in 3D space. This means that the colour of short arrows may indicate a high magnitude, because the vector may point into or out of the plane. The size and number of vector arrows can be adjusted using the Vectors option from the plot context menu.

Figure 5.13: 2D cut plane plot and controls in the properties sidebar.
5.3 Plot types

5.3.3.1 Appearance

In the Appearance dialogue in the context menu, the mapping of colours to data values can be specified for the active plot.

![Appearance dialogue](image)

Key elements of the colour map setup dialogue:

- Variable selection: A separate colour map can be specified for each variable displayed in the active plot.

- Value range: Specifies the range to be mapped into colour space:
  - *Auto Simulation* uses the range from the minimum to the maximum value for the chosen variable across all time steps.
  - *Auto Timestep* is preset to the value range for the current time step.
  - *Fixed* allows you to specify a range manually.
  - *LFL/UFL* sets the value range based on the lower and upper flammability limits for a given variable. This option is only valid for a limited number of variables, including but not limited to FMOLE, FUEL and ER. It will be automatically hidden for variables that do not support it.
  - *Custom* lets you specify a list of values for the colour interval limits.

- Gradient: The drop-down list allows to select the colour sequence that is mapped to the value range.
• Levels: The levels slider specifies the number of discrete colour values when using separate colour steps, and also the number of contours when contours are enabled.

• Colour: "Continuous" maps the values to a continuous range of colours, while "Step" divides the colour map into discrete colours.

• Contour filling: specifies whether the contours should be filled or not.

• Contour outline: Enables or disables the plotting of iso-contours.

5.3.3.2 Porosity verification

By enabling Verify Porosities in the Tools menu you switch on the porosity verification mode. When this mode is enabled, you will see an information box containing the porosity information for the cell the mouse cursor is positioned in. Note that the information is given as blockages rather than porosities. A value of 100% blockage means that no flow can pass through the surface or volume (0% porosity); a fully open passage has 0% blockage (100% porosity). The information comprises the volume blockage of the cell and the area blockages of the two faces perpendicular to each of the three coordinate directions.

Figure 5.15: Blockage information displayed in verification mode.

5.3.4 3D plot

The 3D plot type shows a three-dimensional representation of the geometry, together with simulation results. For scalar variables, four visualisation modes are available:

• Volume: the data is visualised in the form of a “cloud”.

• Surface: the data is visualised as filled contours on the geometry.

• Isosurface: semi-transparent surfaces indicate the locations where the variable assumes specific values.

• 2D slice: the data is visualised as filled contours on axis-aligned planes.

For vector-valued variables, the following visualisation modes are provided:

• Glyphs: the direction and magnitude of vectors are displayed using arrows.

• Streamlines: curves tangent to the vector field convey the direction of the flow.

• 2D Slices: axis-aligned planes slicing through the vector field using the line integral convolution (LIC) visualization technique.
Figure 5.16: A 3D plot with both surface and volume data and streamlines.

Figure 5.17: Visualizing a vector field using the 2D slice with the line integral convolution (LIC) technique.
Warning:

The 3D plot should be used with caution, and not for extracting data values. Interpolation, numerical transformations, transparency settings, perspective distortion and graphical aliasing effects can lead to wrong values being observed. Use monitor points or panels, or the 2D cut plane plot to obtain more accurate values.

Figure 5.18: Different ways to visualise the velocity vector field; upper left: streamlines along a seeding line, integrated both backwards and forwards in time; upper right: same as before, but only integrated forward in time; lower left: streamlines seeded at random locations, lower right: glyphs/arrows.

5.3.4.1 Data selection

A 3D plot can show many variables simultaneously, but there are limitations on the number of variables for a given visualisation mode: A maximum of two variables can be displayed as volumes. In addition, one variable can be shown per mode for the Surface, 2D slice, Glyphs and Streamlines modes. The Isosurface visualisation has no limitations.

Which variables are displayed in which way is prescribed when opening the data file or in the data selection window in the properties sidebar. Every data set has its own legend and settings and can be handled separately.
5.3 Plot types

5.3.4.2 Appearance

For the 3D plot type, the same kind of settings are accessible in the Appearance dialogue in the context menu as in the 2D plot appearance dialogue. If the activated variable is visualised using surface, 2D slices, glyphs or streamlines, additional options become available.

5.3.4.2.1 Predefined color setups

Some variables have predefined color setups available. These can be selected in the "Color setup" dropdown. Selecting a predefined setup will disable the rest of the options.

Table 5.2: Predefined color setups

<table>
<thead>
<tr>
<th>Setup name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flame</td>
<td>Available for &quot;Temperature&quot; variable. This setup uses additive color blending, and sets a value range and color table to give a flame like appearance.</td>
</tr>
<tr>
<td>Smoke</td>
<td>Available for &quot;MassFractionSoot&quot; and &quot;VolumeFractionSoot&quot; variables. This setup sets a dark color table, volumetric lighting and opacity settings to give a smoke like appearance. For the &quot;VolumeFractionSoot&quot; variable, the opacity set will give a realistic viewing distance, corresponding to the Visibility output variable.</td>
</tr>
</tbody>
</table>

5.3.4.2.2 Transparency settings

An additional appearance setting for 3D plot is the transparency settings in the color map view. This makes it possible to set the transparency of each color in the color map. The transparency is defined as a curve where low values (close to the bottom) are more transparent and high values (close to the top) are more opaque. The transparency curve is used to control the visual appearance of volume visualizations, e.g. to make it possible to see through a gas cloud. The transparency setting affects the transparency of all the visualization types, except the streamline visualization, and the glyphs visualization of vector field for which it controls the size of the glyphs.
If the Manual radio button is active, the transparency curve can be manually drawn in the color map window using the mouse while holding down the left mouse button. If the Smoothstep radio button is active a smooth curve is computed for the defined min-max interval, going from a low to a high value (transparent to opaque).

In addition to the transparency curve in the color map, the overall transparency is controlled by the vertical slider to the right of the color map view. The value of this slider is multiplied with the value from the transparency curve, and makes it possible to easily set the general transparency level.

5.3.4.2.3 Surface appearance options

Although the default 3D surface plot algorithm is relatively robust it may show inaccurate results in a few rare instances. Because of this two additional options are available when using surface plotting: Fast interpolation and Offset interpolation.

The Fast interpolation option applies only to 3D surface variables, not to volume variables. By default, the Fast interpolation option is disabled, which means that a more advanced interpolation algorithm for plotting surface values in the 3D plot is activated. The advanced algorithm takes into account porosities as well as grid alignment of the walls; in this way, the visualisation takes into account how the FLACS-CFD solver deals with such objects. The result is that the walls show more correct values on both sides. Currently, the advanced algorithm does not work for left difference primitives and it may fail on older graphics cards. Although the newer algorithm is robust and accurate, in a few rare cases (i.e. for walls aligned exactly with the center of a cell or sub-grid objects near walls) it might still show inaccurate results. It is therefore important to verify the 3D surface values against 2D cut plane plots and/or monitor points and panel values. When the Fast interpolation option is enabled, it does not take into account any porosities or the geometry position in the grid. Although slightly quicker, this can lead to wrong values being shown on the geometry.
Notably, thin walls will show the same value on both sides and the value will be an interpolation between the data on both sides of the wall. For instance, if there is a 2 bar overpressure on one side of the wall and 0 bar overpressure on the other side, the 3D surface plot may show the overpressure as being 1 bar on the wall. The Offset Interpolation option applies only to 3D surface visualization, i.e. projecting cell data onto object or terrain surfaces. If the surface of a large object (like a fuel storage tank) or terrain is cutting through a blocked cell with low porosity, the mapping of variable data on the surface can result in a suboptimal visual quality. E.g. on a curved or sloping surface the intersection may shift between blocked cells with no variable data, and (partly) open cells with variable data, resulting in a very irregular surface visualization. By enabling Offset Interpolation the interpolation algorithm will use variable data from the closest neighbour cell in the direction of the surface normal. The Volume porosity threshold slider / numeric field is used to adjust when the interpolation algorithm shall sample variable data from neighbour cell. The value represents the porosity value of the current intersected cell.

![Figure 5.22: Appearance dialogue section for 2D slices.](image)

**5.3.4.2.4 2D Slice appearance options** A 2D slice plane can be added by clicking on the button “Add plane”. Once a plane has been created, its direction can be chosen using the dropdown menu on the left-hand side. The position can be adjusted using the spin-box, the slider or using the left mouse button while holding down CTRL and Shift. Several planes can be added, with different directions and positions. A plane can be removed by clicking on the “-” button in the rightmost column.

![Figure 5.23: Appearance dialogue section for glyphs.](image)

**5.3.4.2.5 Glyph appearance options** The number and the reference size of the glyphs can be controlled using the two spin-boxes in the “Glyphs” frame. Notice that the size and the color of a glyph depend on the magnitude of the vector at the glyph’s location. In particular, the color of the glyph is given by the current colour map, while the size is determined by the transparency channel. In the picture above, you can see how the size of the glyphs is affected by modifying the transparency curve (in the red box).
5.3.4.2.6 Streamline appearance options  The number, radius and integration time (length) of the streamlines can be set in the dialogue. In addition, you can choose whether the integration should be backward in time, forward in time, or both.

For the seeding type Line, a magenta line segment can be positioned by moving the line's gray end points (green arrow) with the mouse while pressing the CTRL-key. To simplify positioning with the mouse, it is recommended to disable the perspective view and use axis-aligned view directions (red arrow). Alternatively, the exact coordinates of the end points can be defined using the controls available in the Appearance dialogue (red box). By clicking on the lock icons next to the axes names, the coordinates of the two end points are constrained to be the same along that axis. This facilitates the definition of axis aligned seeding lines. For instance, by “locking” the Y and Z coordinates, the seeding line will be parallel to the X axis. The locking is ignored when dragging the end points with the mouse.

For the seeding type Random, the origins of the streamlines are distributed randomly in the domain. It is possible to add several sets of streamline origins (Line or Random). This is particularly useful when several seeding lines are used to show the flow field in different parts of the domain. The parameters of each seeding line can be set individually by highlighting the it in the selection list at the top of the Streamlines dialogue.
5.3 Plot types

5.3.4.2.7 Volume appearance options  When using the volume visualisation you have three options: volumetric, isosurface and maximum value. These three choices are mutually exclusive and represent different ways of visualising volume data. In *Volumetric* mode, a semitransparent cloud is shown, where the colour is determined by accumulating values along rays from the camera position into the data volume. *Maximum value* does the same, but the colour is determined by the maximum value for the given variable in the simulation results along the view direction. *Isosurface* shows an opaque surface representing the data points in the given value range. The accuracy of the volume rendering can be specified by using the slider.

5.3.4.3 3D plot options

![Options window](image)

Figure 5.26: The options window in the properties sidebar for a 3D plot.

5.3.4.3.1 Show legend in sidebar  The legend of this plot is displayed in a dedicated area on the right side of the screen. See also the section regarding the Sidebar legend.

5.3.4.3.2 Grid  When enabling this option, the grid will be shown at the domain boundaries, as long as it does not obstruct the view onto the domain.
5.3.4.3.3 **Anti-aliasing**  
Aliasing is a sampling effect in pixel graphics that leads to distortion or noise in the screen representation. Flowvis includes a special smoothing algorithm for edges to reduce aliasing effects and enhance graphics quality. If there is text in the 3D plot (e.g. monitor point labels) then it may become slightly blurry due to anti-aliasing. This does not affect the legend or picture text.

5.3.4.3.4 **Ambient occlusion**  
Ambient occlusion enhances 3D plots by including radiation from geometry surfaces. It yields a softer appearance than just direct lighting, approximately in the way objects appear on an overcast day.

5.3.4.3.5 **Edge enhancement**  
When edge enhancement is enabled, the edges of geometry objects will be drawn with an extra black line, making it easier to discern where surfaces meet.

![Figure 5.27: Visualisation of the effects of ambient occlusion and edge enhancement.](image)

5.3.4.3.6 **Shadows**  
The shadows cast by the geometry will be rendered if this option is enabled.

5.3.4.3.7 **Sky**  
This option specifies whether a sky should be shown instead of the standard white background above the ground level.

5.3.4.3.8 **Show geometry**  
This option decides whether the geometry should be shown or hidden.

5.3.4.3.9 **Grayscale geometry**  
The geometry can either be shown with the colours defined in CASD, or as gray scale, which makes it easier to observe volumetric data when no surface data is shown.

5.3.4.3.10 **Textured Geometry**  
Enables textured geometry.

5.3.4.3.11 **Realistic lighting**  
Enables realistic lighting.

5.3.4.3.12 **Use Reflections**  
Enables reflections.
5.3 Plot types

5.3.4.3.13 Monitor points  When enabled monitor points will be visualised as spheres. In Flowvis, monitor points are coloured based on their validity given the scenario's geometry, grid and domain:

- **Green**: The monitor point is positioned in a cell that is essentially open, and that does not fulfill any of the criteria for being colored red or yellow.

- **Yellow**: The monitor point is positioned in a cell where both the area blockage and the volume blockage are between 50% and 90%.

- **Red**: The monitor point is positioned in a cell where one or more of the following criteria are fulfilled:
  - Cell volume blockage is greater than 90%.
  - All the cell faces have a porosity blockage greater than 50%.
  - The monitor point is placed outside the domain.

Note:
The colouring of monitor points is not functional when using CGNS files.

5.3.4.3.14 Show Leaks  When enabled the leaks will be visualised.

5.3.4.3.15 Disable Monitor Labels  When enabled the monitor labels will not be shown.

5.3.4.3.16 Panels  When enabled pressure relief panels will be visualized. They are visualized as yellow plane surfaces with the panel name written on top. If a panel yields during the simulation, due to pressure exceeding the opening limit, the color of the panel will gradually become transparent in order to visualize the change in porosity.

Two of the panel types, **hinged rigid** and **popout rigid**, has additional visualization features. The position of the **popout rigid** panel will move when the panel yields, illustrating the "popout"-functionality. The **hinged rigid** panel will indicate the hinge side of the panel using a solid yellow cylinder, and the panels representation will also swing open like a hinged door when the panel yields.

Figure 5.28: Comparing the visualization of the popout-rigid (Panel2), popout(Panel3), hinged-rigid (Panel4) and hinged (Panel 5), before (left image) and after (right image) the panels yield.

5.3.4.3.17 Set panel transparency from yield  When enabled the panels will receive their transparency based on their yield. When disabled the pressure relief panels will receive a fixed transparency (50%) irregardless of the yield value.
5.3.4.3.18 **Show line monitors**  When enabled line monitors will be visualised.

5.3.4.3.19 **Ignition location**  When enabled the ignition location will be visualised.

5.3.4.3.20 **CAD model**  By default, Flowvis shows the geometry used by the Flacs simulator, which consists of axis-aligned shapes of certain kinds. By enabling the CAD model checkbox it is possible to instead see the original CAD geometry. Note that the solution computed by Flacs is defined on a rectangular grid, which may not fit completely with the non-grid-aligned CAD geometry.

![Figure 5.29: The original CAD model shown in Flowvis.](image)

5.3.4.3.21 **Domain clip planes**  When enabled the effect of the Domain settings is activated. The Domain is basically a clipping box that makes it possible to hide parts of the scenario by adjusting the position of the sides in the box. Parts of the scenario outside of the box is not visible when this setting is enabled.

5.3.4.3.22 **Picture text box**  In the lower left corner of each plot, an information summary containing, for example, the job number, the displayed variable(s), and the time is displayed. It may be hidden by disabling the picture text check-box. The box can also be moved in the plot area by dragging it with the mouse.

5.3.4.3.23 **Show title**  In the picture text box (see above) optionally the title assigned in the Scenario settings section of the CASD scenario menu (or in the cs-file) can be displayed.

5.3.4.3.24 **Show description**  The content of the picture text box (see above) can also be expanded by the description specified in the Scenario settings section of the CASD scenario menu (or in the cs-file).

5.3.4.3.25 **Light position**  The Light position setting in the options window sets the direction the light is coming from. It can be a fixed direction, for example South-East. Alternatively, there is a headlight setting, which positions the light source at the viewpoint. The light will cast shadows according to the source direction, with the exception of the headlight setting, which does not yield shadows. The light direction can be set more accurately by holding the CTRL or SHIFT key while scrolling the mouse wheel; pressing CTRL while scrolling sets the elevation, holding SHIFT and scrolling sets the azimuth.
5.3 Plot types

5.3.4.3.26 Ground  The Ground setting adds a ground plane to confine the geometry in the visualisation. The ground and its type neither result from input settings nor do they affect the visualisation of simulation results. There are three ground types to choose from: water, grass, and concrete. In addition, there is the option to turn off the ground plane in the visualisation. The height (Z-coordinate) of the ground plane can be set with the Ground height setting.

![Figure 5.30: Setting the light position.](image)

![Figure 5.31: Example of concrete ground displayed under a geometry.](image)

Note:

In case there is a ground object in the FLACS-CFD geometry, it is recommended to add a small vertical offset (e.g. +0.05m) to the ground height defined in Flowvis to avoid or reduce rendering artefacts. Alternatively you can apply transparency to the ground object.

5.3.4.4 Visualization of pressure relief panels

5.3.4.5 Geometry

The geometry window allows to change the appearance of the geometry objects in 3D plots. Such changes are based on the hue value assigned in CASD or when importing the geometry. Based on the hue, objects can be displayed with a different colour, as mapped by a colour map, and they can be made partially transparent.
Figure 5.32: Different settings for the geometry visualisation. The Geometry window for each subplot is shown side by side with the plot. Top left: standard settings, no transparency and the objects' hue values are directly used as colours; top right: a different colour map has been applied; lower left: the outer walls have been made transparent, the other geometry is fully opaque; lower right: all objects are partly transparent.

To be able to easily make, for example, the outer walls transparent, these parts of the geometry have to be assigned a hue value of their own. Transparency is a powerful tool to visualise solution values, for example a gas cloud, along with the relevant geometry. On complex geometries, however, the use of transparency is, computationally demanding and the performance of Flowvis can degrade.

5.3.4.6 Monitor selection

When very many monitor points or panels are defined in a FLACS-CFD scenario, it may be overwhelming to show them all in a 3D plot. On the other hand, one often wants to illustrate where in space certain measurements, e.g. pressure-time curves shown in a separate plot, are taken. This is possible by defining the monitors that are to be shown as a group, or with a distinct name prefix, in CASD. By right-clicking in the 3D plot and choosing Monitor selection you open a window with the list of monitors defined for the scenario. You can either filter the list by a string, which is useful if you have given the monitors to be shown a common name prefix, or alternatively select the monitors to be shown with the mouse; the latter also works easily with groups of monitors as defined in CASD.
5.3 Plot types

5.3.4.7 Flight mode

Flowvis has an experimental facility to define sequences of viewpoints so that movies can be created in which the camera moves around the geometry as time advances. Flight mode is enabled by clicking the airplane logo in the icon bar of a 3D plot window. This will open the flight path editor.

To define a flight path, follow these steps:

1. Adjust the time slider to the time that should be shown.
2.Arrange the view as it should be at the given time.
3. Click Add in the flight path editor. For the next point, start over from 1.
While the flight path editor is open, the flight path is shown by a green line with gray dots at the defining positions. The flight path can be modified by dragging the gray dots with the mouse while holding down the Ctrl key. Alternatively, you can double-click a viewpoint to jump to it, change the view as desired, and then click Add to update the viewpoint.

This visualisation of the flight path is hidden when the flight path editor is closed. Note that closing the flight path editor does not leave flight mode, so that it is possible to show and export the flight sequence without the flight path being visible. The solution variable(s) shown will also update according to the defined time. When flight mode is deactivated in the icon bar, the viewpoint is set to the current one and hitting the play button will only make the solution advance, without camera movement. The flight path for a 3D plot is saved in the Flowvis presentation file and recalled when re-entering flight mode.

5.4 Selected topics

5.4.1 Filter Variables

Filter variables wraps existing variables and applies a function on the data before returning it. They can be used to calculate, for example, the moving average of a specified variable. The filter variable menu can be found in the context menu of the variables in the Data Selection section.

After selecting a filter it will prompt you for the necessary input required to make it, and then a new variable will be created with the provided name. This new filter variable can now be used as any other variable in the presentation. The Moving Average filter calculates the moving average of a variable over a specified window. The window specified is in seconds and ends at the current timestep. The Maximum over time filter can be used to propagate the maximum value in the cells over time, and the Maximum over axis filter can be used to propagate the maximum value in the cells in the specified direction.

Note:

The "Maximum over axis" filter is only applicable to 3D variables.

5.4.2 Export graphics

Shortcut: Ctrl-E

The File → Export Graphics menu item opens a dialogue that allows to save pages and presentations as pixel graphics (file formats: png, jpg, tif) or vector graphics (file formats: svg, eps, pdf). The file format can be changed by clicking on Browse or by manually changing the extension of the filename.

If multiple pages or time steps are selected for export, Flowvis will put these into a single file if the file format permits that, or alternatively into several files using counters in the file name (%P or %T for pages and time steps, respectively).

When exporting as a pdf document, all time steps for a given page are placed in the same file. Therefore, using the %T counter with pdf documents does not result in multiple documents.

The Specified timesteps option is only available when time is not synchronized. Vice versa, the Time range option is only available when time is synchronized.
5.4 Selected topics

Figure 5.36: The Export graphics dialogue allows to save pages and presentations as pixel or vector graphics.

5.4.2.1 Export to GeoTIFF

If the file format is set to tif, Flowvis will generate a georeferenced image file (GeoTIFF) by including relevant coordinate transformations within the image file. The GeoTIFF parameters group in the Export Graphics GUI will be enabled if the output filename ends with ".tif", and can be used to place the exported image in the correct location when imported into a GIS system. The X and Y offset parameters in the export dialog will be added to the position in the FLACS-CFD coordinate system. If the geometry has been rotated in CASD e.g. in order to align the orientation of objects with the coordinate axis, the Rotate GeoTIFF parameter can be used to correct for this rotation in the export, by applying a negative rotation of the same amount. Values for Offset and Rotation will be initialized using information read from the geometry (.geo) file.

GeoTIFF file export works best with 2D cut plane plots that are oriented in the XY plane. In order to export GeoTIFF files from a 3D plot, the view orientation has to be top-down (use the View Along Z/-Z axis button in the lower right), and the view has to be orthographic (use the Perspective camera icon/button in the lower right).
5.4.2.2 Batch export

By checking the Do batch export option and selecting a number of similar scenarios before clicking on the OK button, all the selected scenarios will be exported using the same settings (e.g. view point, time-step, resolution, etc.)

Note:
When using the batch export option all the selected scenarios should have similar properties.

5.4.3 Export movie

Shortcut: CTRL-SHIFT-E

The File → Export movie menu item opens a dialogue that allows to save pages as movies (file formats: AVI, wmv, MPEG, MPEG4). The export format can be changed by clicking on Browse. The Time range option is only available when time is synchronized. The viewpoint and view direction during movie export can be animated using the flight mode feature.

If you want to export movies of multiple similar scenarios using the same settings, the batch export functionality can be used, in the same way as described for exporting graphics.

![Export Movie Dialogue](image)

Figure 5.37: The Export movie dialogue allows to save pages as movies.

If the option 360 video export is checked, a 360 degree video (using equirectangular projection) will be created from the current camera position.
5.4 Selected topics

5.4.4 Export 360 FOV Graphics

Shortcut: CTRL-ALT-E

The File → Export 360 FOV Graphics menu item opens a dialogue that allows the user to export all pages or the current page to 6 images per page that can be converted to a 360° view using tools such as Marzipano. Each page are exported using the page name postfixed with a single letter representing the camera direction.

![Figure 5.39: The Export 360 FOV Graphics dialogue.](image)

When exporting 360° views the width and height resolution must be the same. This is ensured by locking the two input fields such that it is not possible to enter different values.

Since the purpose of this export type is to generate a 360° view any 2D object in the plot, such as the FLACS-CFD logo overlay, legend or any text box, should only be visible in 1 of the 6 images that are
exported. To facilitate this the export dialogue allows the user to choose in which direction any 2D items should be visible with the *Legend direction* option.

### 5.4.5 Number formats

The formatting of decimal numbers in the legends in Flowvis can be changed in the settings accessible via the *Options* menu, selecting *Preferences*. The number of decimal or significant digits can be set and either scientific or standard notation selected.

![Figure 5.40: Different number format settings and how they affect a sample legend.](image)

### 5.4.6 Terrain in Flowvis

Flowvis will display the terrain in both 2D cut plane plots and 3D plots. You can enable/disable showing the terrain using the *Terrain* checkbox in the plot context menu (right click on the plot). In a 2D cut plane plot, the terrain intersection is drawn as brown dashed line in XZ and YZ planes. In XY planes, brown hachure indicates the areas where the current cut plane is located under the terrain.

![Figure 5.41: Visualisation of terrain in 2D plots with different cut planes and a 3D plot.](image)
5.4 Selected topics

Note:

When using large terrains in Flowvis, the parts that are far away (more than ca. 10 km) from the view point may be clipped. The same clipping (far plane culling) applies to other geometry as well. Zooming in or changing the view angle may help to show the entire geometry, but the best way to avoid the effect is to restrict the range of the terrain during the terrain import in CASD.

5.4.7 Plot linking

Plot linking allows you to connect two or more plots such that they share certain properties. The set of properties that can be linked depends on the plot types that are linked: each plot type has a set of linkable properties and it will only be possible to link the properties that are common to all linked plots. The following properties can be linked:

Table 5.3: Linkable plot properties.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Applicable plot types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables</td>
<td>selected variables</td>
<td>all</td>
</tr>
<tr>
<td>Plot domain X</td>
<td>domain of the plot in the X-direction</td>
<td>2D cutplane plot, 3D plot</td>
</tr>
<tr>
<td>Plot domain Y</td>
<td>domain of the plot in the Y-direction</td>
<td>2D cutplane plot, 3D plot</td>
</tr>
<tr>
<td>Plot domain Z</td>
<td>domain of the plot in the Z-direction</td>
<td>2D cutplane plot, 3D plot</td>
</tr>
<tr>
<td>Plot domain horizontal</td>
<td>horizontal plot domain (current X and Y axis)</td>
<td>scalar time plot</td>
</tr>
<tr>
<td>Plot domain vertical</td>
<td>vertical plot domain</td>
<td>scalar time plot</td>
</tr>
<tr>
<td>Data files</td>
<td>simulations forming the basis for the plots</td>
<td>all</td>
</tr>
<tr>
<td>Colormap settings</td>
<td>range, subdivision and palette of the colormap.</td>
<td>2D cutplane plot, 3D plot</td>
</tr>
<tr>
<td></td>
<td>These settings can be found in the Appearance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>menu.</td>
<td></td>
</tr>
<tr>
<td>3D view</td>
<td>view position and angle, zoom</td>
<td>3D plot</td>
</tr>
</tbody>
</table>

A link can be created by holding the CTRL key down while selecting two or more plots with the left mouse button. After multiple plots have been selected, they are linked by right clicking on one of the highlighted plots and selecting Create link. This will automatically open the Link properties window in the Properties sidebar as described in The properties sidebar. Links can also be created and deleted directly from the Link properties window. In the case that this window is closed, it can be opened by following the instructions in The properties sidebar.
Note:

Only plots on the same page can be linked. A plot can only be part of one link at a time.

Figure 5.42: A link of the 3D view property has been created between the four 3D plots. This ensures that the view settings are always the same for all four plots.

5.4.8 Sidebar legend

The legend of one or several plots can be placed to the right of the plots in a sidebar. There are several use cases for this, the most common ones are:

- several plots on the same page have the same legend,

- it is difficult to place the legend inside the boundaries of the plot.

The legend for a plot can be placed in the sidebar by right clicking on the plot and selecting Show legend in sidebar. The sidebar legend window can be resized horizontally by dragging the gray line between the plots and the right sidebar.

When placing a legend in the sidebar to avoid legend duplication, it may be necessary to disable the legend on other plots. For the 3D plot and 2D cutplane plot the legend can be disabled in the Appearance menu in the plot properties sidebar. For scalar time and scalar line plots, the legend can be disabled from the context menu under Legend location.
5.4 Selected topics

5.4.8.1 Sidebar legend example

This example shows some results from example 3 of the FLACS-CFD-I course.

The two plots shown in figure above have identical legends. One of the legends can be placed in the sidebar.
by right clicking and choosing *Show legend in sidebar*. In this example the legend for the upper plot is placed in the sidebar. The legend for the lower plot can then be hidden by unchecking the *Show legend* option under *Appearance* in the properties sidebar.

![Image of Flowvis tool bar and legend](image-url)

**Figure 5.45**: The same page as before but with the legend placed in the sidebar.

### 5.4.9 Aggregating results

The aggregation dialogue is opened by clicking on the corresponding button in the *Flowvis tool bar*:

![Image of Aggregation button](image-url)

**Figure 5.46**: The *Aggregation* button (red frame) in the Flowvis tool bar.

In some cases it can be beneficial to aggregate results from multiple runs into a single result. For example, you may want to show the maximum overpressure from several explosion simulations as a single 3D field. This can be accomplished by using the *Aggregation* functionality in Flowvis.

The aggregation function always aggregates across time and runs.
5.4 Selected topics

5.4.9.1 Aggregation dialogue

There are a number of different options available in the aggregation dialogue. By default only the most common options are shown. All options can be shown by clicking on "Advanced settings". See below for an overview of the different options available in the aggregation dialogue.

Table 5.4: Aggregation options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Runs</td>
<td>This field lists the runs which can be aggregated. Multiple runs are selected by holding down the CTRL-key and left-clicking the job numbers.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Variables</td>
<td>Here you can select the variable to aggregate. It is only possible to aggregate one variable at a time. All variables that exist in at least one run are displayed.</td>
</tr>
<tr>
<td>Aggregation type</td>
<td>The type of aggregation to perform.</td>
</tr>
<tr>
<td>Save directory</td>
<td>The directory where the aggregated results should be saved.</td>
</tr>
<tr>
<td>Save file name/jobnumber</td>
<td>Job number to assign to the aggregated data when saving it. The standard FLACS-CFD job number format applies here.</td>
</tr>
<tr>
<td>Resample to grid</td>
<td>The aggregation algorithm calculates results on a single grid. This can be the same grid as one of the simulations or completely different. In most cases the results from one or more simulations will need to be resampled to fit this grid. By default the grid of the first simulation is used. In many cases, especially for dispersion simulations, it will be desirable to use a course or more uniform grid for the aggregation.</td>
</tr>
<tr>
<td>Open when finished</td>
<td>When enabled, the aggregated results will be opened in the chosen plot type when finished.</td>
</tr>
<tr>
<td>Custom title</td>
<td>Custom title used in the generated cs-file.</td>
</tr>
<tr>
<td>Custom description</td>
<td>Custom description used in the generated cs-file.</td>
</tr>
<tr>
<td>Behaviour if grid is different</td>
<td>If the selected runs have different grids, you can choose between three courses of action: resampling to a base grid, skipping the simulation, or stopping the aggregation. In most cases resampling to a base grid is the best choice.</td>
</tr>
<tr>
<td>Timesteps</td>
<td>Allows you to specify whether the aggregation should be applied to all timesteps or only to the last timestep. The latter can be useful when aggregating variables such as PMAX.</td>
</tr>
<tr>
<td>Propagate boundary values</td>
<td>When this option is enabled and the grid that the results are resampled to are larger than the original grid for a given run, the boundary values of that run will be propagated/extended outside of the original grid dimensions. This is the conservative and safe option, but may produce artifacts if any of the border values are significant.</td>
</tr>
<tr>
<td>Outside boundary value</td>
<td>This field is only editable when Propagate boundary values is disabled. Allows you to manually specify what value should be assigned to the cells outside the original grid dimensions.</td>
</tr>
</tbody>
</table>

### 5.4.9.2 Aggregation example

The following example demonstrates the aggregation functionality for a simple case where the maximum pressure PMAX is maximised over the results from four scenarios. The runs are based on example 4 from the FLACS-CFD course. Since the PMAX field contains the maximum pressure that has occurred per grid cell in the course of a simulation, it is sufficient to aggregate over the last time step from each of the four simulations.
5.4 Selected topics

Figure 5.49: The settings applied in the Aggregation dialogue for the example with four different scenarios.

Figure 5.50: The P\textsubscript{MAX} result for each of the four scenarios individually.
5.5 Known issues in Flowvis

The current section gives an idea of some deficiencies and the future development of Flowvis, and should help to avoid communication and feedback regarding issues that are already known. If you find other issues, limitations or bugs please contact flacs@gexcon.com and help us to improve Flowvis and make in the way you want it to be!

- OpenGL warnings may be displayed when creating a new 3D plot.
- Dynamic GPU switching may lead to crashes in presentations with 3D plots on machines with both an Intel integrated GPU and separate NVIDIA GPU. if experiencing such problems, try to disable dynamic GPU switching (Optimus) and fix graphics processing to the NVIDIA device (in the Windows Device Manager, NVIDIA control panel, or BIOS).
- For 3D plots of large data sets, the time slider may not cover the whole simulated time as long as not all data has been loaded.
- Although the default 3D surface plot algorithm is relatively robust it may show inaccurate results in a few rare instances (i.e. for walls aligned exactly with the center of a cell or sub-grid objects near walls). It is therefore important to verify 3D surface values against 2D cut plane plots and/or monitor points and panel values.
- In 1D-plots the domain selection slider and plot can go out of sync (pressing the reload button should correct the plot, but will reset the axis ranges to the initial values).
- When opening the Figure options dialogue for a 1D-plot, the plot legend may get messed up with extra data; workaround: press the reload button.
- Sometimes plots using logarithmic scale are displayed wrongly (grey areas).
- Switching language to Chinese may lead to some characters being represented as rectangles. This can be fixed by switching to a different font-family in the Settings dialog opened from the menu "Options -> Preferences...-> Fonts".
5.6 Recommended settings in Flowvis

For pages with several plots the following settings in the Options menu may help to work efficiently and obtain optimal results:

**Synchronize Time** (default)

**Display seconds** changes the time unit displayed in the plot from milliseconds to seconds; this is recommended especially for longer simulations such as dispersion and fire scenarios.

5.7 Generating a log file in Flowvis

Flowvis will write messages regarding warnings and errors to the console if needed. The detail level of the output can be changed using the `-v` or `--verbose` commandline parameter, or by setting the `FLOWVIS_LOG_LEVEL` environment variable. Starting Flowvis with `-v` will increase the log level to include info messages. Starting Flowvis with `-vv` or `-v -v` (repeating it two times), will increase the log level to include both info and debug messages. The same functionality can be achieved by setting the `FLOWVIS_LOG_LEVEL` environment variable to 1 or 2.

It is also possible to make Flowvis create a log file. In order to trigger the generation of a log file, use the commandline parameter `-l` or `--logfile`, followed by the full path to the logfile, e.g.

```
> flowvis.exe -l C:/tmp/flowvis-log.txt
```

An alternative way to create a log file is to set the environment variable `FLOWVIS_LOG_FILE` to a file path. Please note that the directory in the path must exist or Flowvis will generate an error message and fail to start.
Chapter 6

Utility programs in FLACS-CFD

This chapter describes the utility programs in FLACS-CFD. You can run these programs from the command line in the terminal window under Linux, or on the command line in the cmd window under Windows. It is possible to start the cmd command shell from the FLACS-CFD RunManager. The utility programs are categorised according to application:

1. **Geometry, database, grid, and porosities**: geo2flacs, convertdb, gm and FGC (Flacs Geometry Calculator)
2. **Release source modelling**: Jet, Jet2, Flash
3. **Modifying simulation files**: cofile (deprecated), cofile2, comerge autostop
4. **Post-processing legacy simulation output**: r1file, r3file, a1file, a3file
5. **CGNS tools**: CGNS tools

**Note:**
Some of the functionality covered by the utilities have been replaced by the FLACS-CFD Python API. Users are encouraged to use the FLACS-CFD Python API in such cases.

6.1 **Geometry, grid and porosities**

6.1.1 **geo2flacs**

The geo2flacs utility allows to import geometry from various 3D file formats to FLACS-CFD. Feedback on geo2flacs is very much appreciated and should be directed to flacs@gexcon.com. Please note especially the sections on geometry verification and geometry import problems and solutions.

6.1.1.1 **File formats**

6.1.1.1.1 **geo2flacs input file formats** The geo2flacs utility supports the following input formats:

- AVEVA® RVM files (PDMS review/exchange file format)
- Microstation dgn versions 7 and 8
- AutoCAD 2013 and 2018 (as DWG/DXF or when contained in a dgn file)
- STL, OBJ, IFC/STEP, FBX, 3DS, LWO and DAE are supported through the Assimp library. For more information about Assimp see their web page.
In addition to the dgn format geo2flacs can import from the prp format (Propagated steel, Frameworks format), which is very similar to the dgn format. The files are handled in the same way as dgn files. The dgn files of a project are often accompanied by a drv file (Design Review). The information in the drv file can be used by geo2flacs to group single primitives into larger objects and to give the objects the proper names.

Files in dgn format are the native output of Microstation, but also other software packages can produce dgn files. Notably, PDMS geometries can be exported in dgn format using the PDMS export driver ExPLANT-I.

Remarks:

The following dgn element combinations are known to work with geo2flacs:

- shape-line combinations (boxes),
- line/line string (boxes, cable trays),
- cone,
- arc/ellipse-line combination (cones/cylinders),
- line (AutoCAD smart solid boxes),

6.1.1.2 Output format

The output format of geo2flacs can be one of the following:

- CASD DB2 geometry database
- CASD GEO geometry file
- CASD CO geometry file

6.1.1.3 Manual geometry import with geo2flacs

The current section describes how to steer the geometry import process from the command line, directly using geo2flacs. This can be useful if special options of geo2flacs need to be used or when automating the geometry handling with scripts.

geo2flacs is a command line program using mandatory and optional arguments. It is run from the Linux command line as

> run geo2flacs [options] INPUT-FILES...

and on Windows by:

> geo2flacs [options] INPUT-FILES...

The input files may be a list of files and/or directories. If a directory is specified, all recognised files in the directory and subdirectory will be imported.

6.1.1.4 Command line options

The following command line options are recognised by geo2flacs:

<table>
<thead>
<tr>
<th>Short Option</th>
<th>Long Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a</td>
<td>--noAlign</td>
<td>Do not align objects to the nearest axis and allow irregular objects. This option is not intended for normal FLACS-CFD simulations and should be used with caution. Note the warning about non-orthogonal elements!</td>
</tr>
<tr>
<td>Short Option</td>
<td>Long Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>–cofile=out.dat3</td>
<td>Create CO geometry file from input files.</td>
<td></td>
</tr>
<tr>
<td>–geofile=out.geo</td>
<td>Create GEO geometry file from input files.</td>
<td></td>
</tr>
<tr>
<td>-d</td>
<td>–minDia=DIAMETER</td>
<td>Do not export objects with diameter smaller than DIAMETER (mm).</td>
</tr>
<tr>
<td>-f</td>
<td>–force</td>
<td>Overwrite existing output file.</td>
</tr>
<tr>
<td>-l</td>
<td>–min-length=LENGTH</td>
<td>Do not import objects with largest length smaller than LENGTH. (mm)</td>
</tr>
<tr>
<td>-n</td>
<td>–group=NUMBER</td>
<td>Group primitives into objects containing NUMBER primitives. This parameter has no function if a drv file is used or for terrain conversion. The primitives will not be carried over from one input file to the next, thus the last primitives converted from one input file might be converted into an object containing less than NUMBER primitives. In some situations more than NUMBER of primitives might be grouped together.</td>
</tr>
<tr>
<td>-o</td>
<td>–outdir=DIR</td>
<td>Output subdirectory (default = out)</td>
</tr>
<tr>
<td>-r</td>
<td>–rotate=ANGLE,X,Y,Z</td>
<td>Rotate the geometry ANGLE degrees around the z-axis through the point (X,Y,Z) before export.</td>
</tr>
<tr>
<td>-t</td>
<td>–translate=X,Y,Z</td>
<td>Translate the geometry by the vector (X,Y,Z) before export. If one or more of the coordinates are negative when using the translate option the coordinates must be specified like this: -t&quot;-100,-100,-100&quot;. Note the double quotes (&quot;&quot;)) around the coordinates.</td>
</tr>
<tr>
<td>-s</td>
<td>–scale=FACTOR</td>
<td>Scale the geometry by FACTOR before export.</td>
</tr>
<tr>
<td>-v</td>
<td>–verbose</td>
<td>Produce verbose output (default).</td>
</tr>
<tr>
<td>-x</td>
<td>–skip-zero-length</td>
<td>Skip cylinders with zero length, otherwise a small length is added.</td>
</tr>
<tr>
<td>-c</td>
<td>–connect-circles</td>
<td>When enabled it may fix problems where pipes are not imported.</td>
</tr>
<tr>
<td>–enable-curvature-analysis</td>
<td>Enabling this option may improve the import, especially of pipes, if the CAD file contains many mesh objects (objects represented by triangles or polygons).</td>
<td></td>
</tr>
<tr>
<td>–enable-brep-analysis</td>
<td>Experimental feature for left difference objects found in DWG files</td>
<td></td>
</tr>
<tr>
<td>–disable-mesh-fallback</td>
<td>When the fallback is disabled unrecognized objects will be imported as bounding boxes</td>
<td></td>
</tr>
<tr>
<td>–disable-merge-shapes</td>
<td>Disable attempt to merge primitives that constitute a box</td>
<td></td>
</tr>
<tr>
<td>–split-lines</td>
<td>When this option is enabled lines will be split into segments using intersecting faces</td>
<td></td>
</tr>
<tr>
<td>–rvm-insulation</td>
<td>Enabling this option will cause geometry tagged as insulation in the CAD model to also be imported.</td>
<td></td>
</tr>
<tr>
<td>–rvm-obstruction</td>
<td>Enabling this option will cause obstruction primitives to also be imported.</td>
<td></td>
</tr>
<tr>
<td>–flipzy</td>
<td>Flip Z and Y axis.</td>
<td></td>
</tr>
<tr>
<td>–center-xy</td>
<td>Center imported geometry at origin in X and Y direction.</td>
<td></td>
</tr>
<tr>
<td>–center-z</td>
<td>Center imported geometry at origin in Z direction.</td>
<td></td>
</tr>
<tr>
<td>-?</td>
<td>–help</td>
<td>Display the help text.</td>
</tr>
</tbody>
</table>
### 6.1.1.5 Geometry verification

As with many conversions between different file formats, the geo2flacs export cannot be guaranteed to produce entirely correct results. It is therefore very important to do a thorough verification of the geometry in CASD. Because CASD only offers a subset of the functionality in most CAD packages some information may be lost and some types of information may be interpreted incorrectly.

Special attention should be paid to details having significant influence on the simulation results. These include walls, decks (with and without grating) and major obstacles. For gratings, the degree of porosity can have a significant influence on explosions, but no porosity information is exported by geo2flacs, and it is often not even modelled in the upstream CAD software (gratings are often represented as plates). In such cases, porosity information has to be added in CASD to take effect in the FLACS-CFD simulation.

### 6.1.1.6 Importing materials

Materials can be imported from DGN, OBJ, and FBX. As stated in the section about materials, a material can have both a colour and a texture, both of which are supported when importing files using geo2flacs.

#### 6.1.1.6.1 Texture filenames

In some cases the importer will not be able to load all the different texture types (color texture, normal map, roughness map, metallic map). To alleviate this the importer automatically tries to find the missing file(s) based on the filename of the one(s) is has found. It assumes the main part of the filename is identical, and that the names follow one of the supported naming schemes in the table below.

<table>
<thead>
<tr>
<th>Color texture</th>
<th>Normal map</th>
<th>Roughness map</th>
<th>Metalness map</th>
</tr>
</thead>
<tbody>
<tr>
<td>basename_baseColor.png</td>
<td>basename_normal.png</td>
<td>basename_metallic.png</td>
<td>basename_roughness.png</td>
</tr>
<tr>
<td>basename_basecolor.png</td>
<td>basename_normal.png</td>
<td>basename_metallic.png</td>
<td>basename_roughness.png</td>
</tr>
<tr>
<td>basename_Base_Color.png</td>
<td>basename_Normal.png</td>
<td>basename_Metallic.png</td>
<td>basename_Roughness.png</td>
</tr>
<tr>
<td>basename_albedo.png</td>
<td>basename_normal.png</td>
<td>basename_metalness.png</td>
<td>basename_roughness.png</td>
</tr>
<tr>
<td>basename_col.png</td>
<td>basename_nrm.png</td>
<td>basename_met.png</td>
<td>basename_rgh.png</td>
</tr>
</tbody>
</table>

#### 6.1.1.6.2 Texture coordinates for DGN

The DGN format support a variety of different texture coordinate mapping modes. See the table below for which UV types these are mapped to.

---

It is mandatory to specify one of the output file options (–database, –cofile or –geofile).

If several transformations are applied in a geo2flacs call, then the order is first scaling, then translation, finally rotation.

**Note:**

Until FLACS v10.6r3 the order was different (first rotation, then translation, finally scaling), therefore the arguments may need to be changed from FLACS v10.7 to obtain the same result as in previous versions.
Table 6.3: Texture coordinate mapping between DGN and FLACS-CFD

<table>
<thead>
<tr>
<th>DGN mapping mode</th>
<th>FLACS-CFD UV type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parametric</td>
<td>Planar local</td>
</tr>
<tr>
<td>Planar</td>
<td>Planar world</td>
</tr>
<tr>
<td>Elevation Drape</td>
<td>Drape</td>
</tr>
<tr>
<td>Directional Drape</td>
<td>Drape</td>
</tr>
<tr>
<td>Cubic</td>
<td>Cubic</td>
</tr>
<tr>
<td>Spherical</td>
<td>Sphere</td>
</tr>
<tr>
<td>Cylindrical</td>
<td>Cylinder</td>
</tr>
<tr>
<td>Solid</td>
<td>None</td>
</tr>
</tbody>
</table>

6.1.1.7 Visualisation and FLACS-CFD model

Geometry that contain polygonal meshes that cannot be automatically converted to a box, cylinder, GTC, CP8, torus or rectangular torus will by default be imported as meshes. In previous versions of FLACS-CFD the algorithm in geo2flacs would create a simpler representation (e.g., a bounding box) of the polygonal mesh in the FLACS-CFD model, but this is no longer default behaviour.

6.1.1.8 Geometry import problems and solutions

6.1.1.8.1 Diagnostic output  geo2flacs will by default output verbose diagnostics. This information can be redirected to a file by using the “>” sign:

Linux:
> run geo2flacs -v -e . > out.txt

Windows:
> geo2flacs -v -e . > out.txt

This will create a file out.txt in the current working directory.

6.1.1.8.2 Note about PDMS ExPLANT-I  There is a bug in ExPLANT-I that sometimes exports geometry in a format that cannot be processed with geo2flacs. If an object is represented by only “Shape” elements it will not be exported.

6.1.2 gm

The gm program can be used to create grids through scripting.

Note:

The functionality provided by the gm utility is now also provided as part of the FLACS-CFD Python API and this should be used in preference to the gm utility.

The following command starts the gm program:

Linux:
> run gm

Windows:
> gm

gm can also take input from a text file. Either of the following methods can be used.

Redirected input (file):
> run gm < file

Piped input (process):
> cat file | run gm
Table 6.4: gm commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>help</td>
<td>show help file (using nedit -read)</td>
</tr>
<tr>
<td>open filename</td>
<td>open and read grid file</td>
</tr>
<tr>
<td>save filename</td>
<td>save grid file</td>
</tr>
<tr>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>add start[:stop[:stride[:n]]]</td>
<td>add grid lines or regions</td>
</tr>
<tr>
<td>delete start[:stop]</td>
<td>delete grid lines or regions</td>
</tr>
<tr>
<td>smooth start:stop</td>
<td>make smooth transition</td>
</tr>
<tr>
<td>stretch start:stop[:factor]</td>
<td>make stretched grid (not implemented)</td>
</tr>
<tr>
<td>refine pos:fac:dia</td>
<td>refine existing grid around a point</td>
</tr>
<tr>
<td>translate value</td>
<td>translate existing grid by given value</td>
</tr>
<tr>
<td>copy x</td>
<td>y</td>
</tr>
<tr>
<td>print</td>
<td>print grid</td>
</tr>
<tr>
<td>format</td>
<td>set print format (default 10f8.3)</td>
</tr>
<tr>
<td>plot</td>
<td>plot grid (using gnuplot)</td>
</tr>
<tr>
<td>small 0.001</td>
<td>set small value</td>
</tr>
<tr>
<td>margin 0</td>
<td>set default margin for smooth and stretch</td>
</tr>
<tr>
<td>opaque 1</td>
<td>set default opaque value for add and copy</td>
</tr>
<tr>
<td>exit</td>
<td>exit program</td>
</tr>
</tbody>
</table>

All grid line values are real, it is not possible to select indices.
The smooth command uses default margin 0, i.e. select inner lines or set margin 1 and select outer lines, see figure below.

```
---------------
| | | | | selected grid lines
---------------
1 0 0 1 margin (0 = inner, 1 = outer)
```

A range given by a single colon means -Inf to +Inf, hence 'delete :' will delete all grid lines.
The refine command works on a predefined grid region, see the figure below.

```
| | | | | | original grid
| | | | | | defined points
| | | | | | removed line 3 and 4
| | | | | | added 4 new lines around p
| | | | | | smooth on both sides
---------------
1 2 3 4 5 6 7 8 9 10 11
```

Table 6.5: Grid refinement parameters

<table>
<thead>
<tr>
<th>Letter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>given centre position, pos</td>
</tr>
<tr>
<td>d</td>
<td>size of CV that contains point p multiplied by given factor, fac</td>
</tr>
<tr>
<td>a</td>
<td>p-d, lower point</td>
</tr>
<tr>
<td>b</td>
<td>p+d, upper point</td>
</tr>
</tbody>
</table>
Refine algorithm:

1. Delete all lines in range a:b; at least 2 grid lines should remain on each side of p for the subsequent smoothing.
2. Add 4 new lines with given distance, dia, centred around p
3. Create smooth grid on both sides

6.1.2.1 Examples

6.1.2.1.1 Create homogeneous grid with 1 m cells

The following sequence of commands creates a homogeneous grid with 1 m cells in all directions over a simulation domain extending from 0 m to 10 m in all directions.

```
gm> x add 0:10:1
gm> y add 0:10:1
gm> z add 0:10:1
gm> save cg010101.dat3
```

The grid is saved in a file named cg010101.dat3. Assuming a 010101-job number this grid can be used in a FLACS-CFD simulation.

6.1.2.1.2 Print grid

The gm program can read the FLACS-CFD cg-file and print the position of the grid lines on the screen:

```
gm> open cg010101.dat3
gm> format 10f8.3
gm> x print
gm> x 11
0.000 1.000 2.000 3.000 4.000 5.000 6.000 7.000 8.000 9.000 10.000
gm> format 5f10.1
gm> y print
gm> y 11
0.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 9.0 10.0
gm> format f0.2
gm> z print
gm> z 11
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
```

The gm print format specification is a Fortran format string with certain modifications and limitations:

- the format string may contain maximally 14 characters (not counting outer parentheses),
- outer parentheses will automatically be added if not present,
- if present, outer parentheses must be balanced,
- no commas, spaces or semicolons are allowed (those are input separators in gm).

Examples of formats for gm print:
Table 6.6: Examples of formats for gm print.

<table>
<thead>
<tr>
<th>format string</th>
<th>notation</th>
<th>columns</th>
<th>field width</th>
<th>decimals</th>
</tr>
</thead>
<tbody>
<tr>
<td>10f8.3</td>
<td>decimal</td>
<td>10</td>
<td>8</td>
<td>3</td>
</tr>
<tr>
<td>5f10.1</td>
<td>decimal</td>
<td>5</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>5e15.7</td>
<td>exponential</td>
<td>5</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>5es15.7</td>
<td>scientific</td>
<td>5</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>5en15.7</td>
<td>engineering</td>
<td>5</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>(10(1x(f0.3)))</td>
<td>decimal</td>
<td>10</td>
<td>minimal</td>
<td>3</td>
</tr>
</tbody>
</table>

If the format field width is too small for a given value then gm will print stars instead of the value (e.g. **********).

6.1.2.1.3 Smooth grid region  Make new grid file cg010102.dat3, smooth region in x direction.

gm> x add -2 0 10:20:1 30 32
gm> smooth 0:10 20:30
gm> y copy x : delete -2 32
gm> z add 0:20:1
gm> save cg010102.dat3
gm> exit

6.1.2.1.4 Refine grid region  Refine grid based on cg010102.dat3, jet direction is x, therefore refining in y and z directions Save as cg010103.dat3.

gm> open cg010102.dat3
gm> y refine 15.5:2.5:0.2
gm> z refine 5.5:2.5:0.2
gm> save cg010103.dat3
gm> exit

6.1.3 FGC (Flacs Geometry Calculator)

FGC can be started from the 'Calculate' option under the 'Porosities' menu in CASD, from RunManager or from the command line (example for job no. 010100):

Linux:
> run fgc 010100

Windows:
> fgc 010100

It is possible to run FGC without employing a graphical user interface (GUI), by the command: Linux:

> run fgc - 010100 [OPTION...]

Windows:
> fgc - 010100 [OPTION...]

This can be practical when running FGC for a batch of simulations at once. Multiple jobs can be started with the command:

Linux:
> run fgc - 010100 020100 [OPTION...]

Windows:
> fgc - 010100 020100 [OPTION...]

Even when FGC is invoked with the option for no GUI, on Linux, certain X libraries should still be installed on the computer being used. FGC supports a number of different command line options when run without GUI:
6.2 Release source modelling

Table 6.7: FGC command line options

<table>
<thead>
<tr>
<th>Short Option</th>
<th>Long Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>n/a</td>
<td>Run FGC without graphical user interface</td>
</tr>
<tr>
<td>-h</td>
<td>–help</td>
<td>Show help message</td>
</tr>
<tr>
<td>-j</td>
<td>–jobs</td>
<td>Job numbers to calculate for. Comma separated</td>
</tr>
<tr>
<td>-l</td>
<td>–lengthscale</td>
<td>Specify whether length scales should be calculated. Default value is 0 (false)</td>
</tr>
<tr>
<td>-n</td>
<td>–numThreads</td>
<td>Maximum number of threads to use. Default is 2</td>
</tr>
<tr>
<td>-p</td>
<td>–porosity</td>
<td>Specify whether porosities should be calculated. Default value is 1 (true)</td>
</tr>
<tr>
<td>-t</td>
<td>–turbulence</td>
<td>Specify whether turbulence factors should be calculated. Default value is 1 (true)</td>
</tr>
<tr>
<td>-g</td>
<td>–geometryModel</td>
<td>Specify which geometry model to use; FLACS or CAD. Default is FLACS</td>
</tr>
<tr>
<td>n/a</td>
<td>–closeGaps</td>
<td>When enabled gaps in area porosities will be attempted closed. Default value is 0 (false). See the FGC technical reference for more information</td>
</tr>
<tr>
<td>-m</td>
<td>–mergeMeshes</td>
<td>When enabled adjacent mesh objects will be merged to form solid objects.</td>
</tr>
<tr>
<td>-v</td>
<td>–version</td>
<td>Prints the version</td>
</tr>
<tr>
<td>-V</td>
<td>–verbose</td>
<td>Sets verbosity. Default is 0. Accepts 0 and 1</td>
</tr>
</tbody>
</table>

A technical description of FGC can be found in Technical Reference.

6.2 Release source modelling

FLACS-CFD includes several tools (Jet, Jet2, Flash) and techniques (e.g., Entrainment modelling for leaks) for modelling releases. These are presented in the following subsections.
The set of species available in the different tools depends on the typical applications and is summarised in the following table.

Table 6.8: Species supported in the Jet and Flash utilities

<table>
<thead>
<tr>
<th>Formula</th>
<th>Species</th>
<th>Jet</th>
<th>Jet2</th>
<th>Flash</th>
<th>Flash2</th>
</tr>
</thead>
<tbody>
<tr>
<td>H2</td>
<td>HYDROGEN</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2</td>
<td>OXYGEN</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O3</td>
<td>OZONE</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2O</td>
<td>WATER</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>H2S</td>
<td>HYDROGEN SULFIDE</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N2</td>
<td>NITROGEN</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>O2+N2</td>
<td>AIR</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>CH4</td>
<td>METHANE</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>CO</td>
<td>CARBON MONOXIDE</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CO2</td>
<td>CARBON DIOXIDE</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2H2</td>
<td>ACETYLENE</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>C2H4</td>
<td>ETHYLENE</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>C2H6</td>
<td>ETHANE</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>C2H3F</td>
<td>FLUORETHYLENE</td>
<td></td>
<td>+</td>
<td>+</td>
<td></td>
</tr>
<tr>
<td>C3H6</td>
<td>PROPYLENE</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>C3H8</td>
<td>PROPANE</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>
6.2.1 Jet

The jet utility program is used to compute the area and the subsonic velocity after shock of a jet issuing from a high-pressure reservoir (note that a newer version of the jet utility program, Jet2, is now available). From a high-pressure reservoir (stagnation point), there is isentropic flow through the nozzle (heat in/out can be specified but the heat transfer coefficients are normally not considered). This is followed by a single normal shock (where Rankine Hugoniot relations are utilised) which is subsequently followed by expansion into ambient air. No air entrainment is considered. The area that is reported in FLACS-CFD is the area of the expanded jet and the reported velocity is the subsonic velocity after the shock.

A schematic of the jet model is shown below:

![Schematic of under-expanded jet model](image)

Figure 6.1: Schematic of under-expanded jet model showing stations where analytical models are applied; sonic conditions assumed at jet exit (station 1); normal shock located at stations 2-3 interface; subsonic flow at station 3.

The jet utility program is started with the following command:

Linux:

```bash
> run jet < jet-in > jet-out
```

Windows:

```bash
> jet < jet-in > jet-out
```
6.2 Release source modelling

Note:

The jet utility program can use two different models to compute the expanded-jet properties:

- the **single planar shock model** (Birch model),
- the **Ewan-Moodie model**.

In short, the Birch model (single planar shock model) assumes that the jet passes through a single planar shock, while the Ewan–Moodie model is based on energy conservation and the assumption that velocity is sonic. The Ewan–Moodie model gives better agreement with experiments in general, irrespective of the simulation type. However, the Ewan–Moodie model results in higher jet velocities and thus simulation requires greater cpu time compared to the planar shock model (Birch model). The Birch model (standard jet model) is the default for dispersion. For fire simulations, the Ewan–Moodie model is recommended as the correct velocity value is essential to predict the flame shape in jet fire simulations. It is possible to activate the Ewan–Moodie model (the Birch model is default) in any type of simulations by running the jet utility program from the command line specifying the command-line option `-ewan` and `moodie` to both the Jet and Jet2 utility programs. Example of dispersion simulation results for a hydrogen jet released from 60 bar reservoir, is shown in plot below, use of Ewan-Moodie model compared to Birch model (standard jet model).

![Centerline Volume Fraction Decay](image)

**Figure 6.2:** Inverse Volume Fraction of fuel as function of axial displacement, dispersion simulation results for a hydrogen jet released from 60 bar reservoir, use of Ewan-Moodie model compared to Birch model (standard jet model)

Below is an example of a basic jet input file (jet-in):

```plaintext
cl-file ! output format (-, *, cl-file, ...).
'METHANE=0.8,ETHANE=0.1,PROPANE=0.04,BUTANE=0.06' ! gas type (AIR, METHANE, ...)
100 ! reservoir volume (m3)
50 30 ! reservoir pressure (barg) and temperature (C)
1 30 ! atmospheric pressure (bara) and temperature (C)
0 0 ! heat transfer coefficients (J/s) and (J/sK)
30 ! wall temperature (C)
0.0355 0.62 ! nozzle diameter (m) and discharge coefficient (-)
0 ! start time (s)
1 10 ! time step (s) and number of iterations (-)
"+XJ" ! leak control string
1e-5 1e5 ! shutoff pressure (barg) and release mass (kg)
0.1 ! relative turbulence intensity RTI (-)
0.1 *D ! turbulence length scale TLS (m) + function
```

FLACS-CFD v22.2 User’s Manual
Most of the parameters can be defined intuitively, a brief description of the more intricate parameters is given below.

The first line determines that the output of the jet utility program will be written in the format of acl-file, the leak file used to define gas releases in FLACS-CFD. This is the most useful output format and is described further below.

**The gas composition** is defined on line 2 using the species names and volume fractions. Table [Species supported in the Jet and Flash utilities](#) shows which species are available with the jet utility program. Alternatively, a component may be specified as GENERIC along with the molecular weight (W) and the ratio of specific heats (KAPPA). For example, H2S can be may be specified as GENERIC (W=34.081, KAPPA=1.331). It is also possible to specify a mixture of several user-defined species, for example:

```
*METHANE=0.8,Generic(W=34.1,KAPPA=1.33)=0.1,Generic(W=24., KAPPA=1.4)=0.1*
```

In the jet utility program, a fixed value of KAPPA is used for each species and assumed to be (approximately) valid over the relevant temperature range. This holds for both pre-defined species (for example METHANE) and user-specified ones (GENERIC).

**The heat transfer coefficients** do not need to be defined, the default zero values can be kept.

**The time step** and the **number of iterations** control the duration and the times at which the properties of the jet will be written in the cl-file. In this example, the properties will be written in the cl-file each second over a duration of 10 s.

**The leak control string** will be directly added into the cl-file for the definition of the direction of the leak. **The turbulence length scale** setting 0.1 * D relates the turbulence length scale to the diameter D of the area. The turbulence length scale value in the output file below results from the circular AREA prescribed as 3.3311 * 10^(-2) m^2, which leads to a diameter D of 0.20594 m, for which 0.1 * D gives a TLS of 2.0594 * 10^(-2) m.

The output file generated by the jet utility program used with the previous inputs is:

```
Enter output format (-, *, cl-file, ...):
Enter gas type (AIR, METHANE, ...):
Enter reservoir volume (m^3):
Enter reservoir pressure (barg) and temperature (C):
Enter atmospheric pressure (bara) and temperature (C):
Enter heat transfer coefficients (J/s) and (J/s/K):
Enter wall temperature (C):
Enter nozzle diameter (m) and discharge coefficient (-):
Enter start time (s):
Enter time step (s) and number of iterations (-):
Enter leak control string:
Enter shutoff pressure (barg) and release mass (kg):
Enter relative turbulence intensity RTI (-):
Enter turbulence length scale TLS (m) + function:
```

```
# GENERAL:
    gas type = METHANE=0.8,ETHANE=0.1,PROPANE=0.04,BUTANE=0.06
    mole weight = 21.093 kg/kmol
    heat ratio, kappa = Cp/Cv = 1.291 -
    critical pressure ratio = 0.547 -

# RESERVOIR:
    critical pressure = 0.827 barg
    pressure = 50.000 barg
    temperature = 30.000 C
    density = 42.679 kg/m^3
    volume = 100.000 m^3
    initial mass = 4267.876 kg

# MACH:
    speed of sound, M=0 = 392.713 m/s
    speed of sound, M=1 = 366.957 m/s
    pressure, M=1 = 26.919 barg
    temperature, M=1 = -8.460 C
    density, M=1 = 26.758 kg/m^3
    maximum velocity = 1030.249 m/s
```

---

**FLACS-CFD v22.2 User’s Manual**
# NOZZLE:
- effective diameter = 27.953 mm
- effective area = 613.675 mm^2
- discharge coefficient = 0.620
- sonic massflow = 6.026 kg/s
- jet force = 3863.166 N

# ATMOSPHERIC:
- pressure = 1.000 bara
- temperature = 30.000 C
- density = 0.837 kg/m^3

The first 14 lines of this file concern the input parameters defined in the jet-in file. The 34 following lines summarise the properties of the jet and physical parameters used in the computations.

The jet force, reported as

jet force = 3863.166 N

above, is the calculated force acting from the jet on the nozzle (i.e., the nozzle from which the jet comes out). For example, if the jet is acting on a barrier plate (or other obstruction) some distance downstream of the leakage point, the reported jet force is in general different from the net force acting on the barrier plate some distance downstream of the leakage point. The force acting on the barrier plate due to the downstream jet, will depend on the distance between the leak point and the barrier plate, the degree of entrainment (mixing with ambient air) before the jet hits the barrier plate, etc.

The last 15 lines constitute the cl-file that will be used by FLACS-CFD for a dispersion simulation. Therefore, if a leak position has been defined in CASD, for a scenario number 999999, this file (with the 48 first lines removed) can be renamed cl999999.n001, and can be used directly for a dispersion simulation. In the new version of the jet utility program, Jet2, specifying the option -silent in the command line, automatically removes the 48 first lines of the output file. The command line is:

**Linux:**

```
> run jet -silent < jet-in > jet-out
```

**Windows:**

```
> jet -silent < jet-in > jet-out
```

Below is an example of a basic jet input file with a generic gas specification (jet-in-generic):

```
cl-file ! Enter output format (-, *, cl-file, ...):
'GENERIC(kappa=1.41,W=2.01594,co_density=130.03)' ! Enter gas type (AIR, METHANE, ...):
0.1 ! Enter reservoir volume (m^3):
800 20 ! Enter reservoir pressure (barg) and temperature (C):
1 20 ! Enter atmospheric pressure (bara) and temperature (C):
0 0 ! Enter heat transfer coefficients (J/s) and (J/s/K):
20 ! Enter wall temperature (C):
0.004 1 ! Enter nozzle diameter (m) and discharge coefficient (-):
0 ! Enter start time (s):
0.25 100 ! Enter time step (s) and number of iterations (-):
'J+X' ! Enter leak control string:
```

FLACS-CFD v22.2 User's Manual
Utility programs in FLACS-CFD

0.001 1000 ! Enter shutoff pressure (barg) and release mass (kg):
0.1 ! Enter relative turbulence intensity RTI (-):
0.1 'D' ! Enter turbulence length scale TLS (m) + function:

The line
'GENERIC(kappa=1.41,W=2.01594,co_density=130.03)' ! Enter gas type (AIR, METHANE, ...):

specifies the exact built-in parameters for hydrogen gas in the jet utility program and is therefore equivalent to

'HYDROGEN' ! Enter gas type (AIR, METHANE, ...):

Make sure to use the option -real_gas_law in order to take real gas effects into account whenever applicable (only validated for hydrogen), or use the option -ideal_gas_law to ensure ideal gas behaviour (default).

Linux:
> run jet -silent -real_gas_law < jet-in-generic > jet-out-generic

Windows:
> jet -silent -real_gas_law < jet-in-generic > jet-out-generic

It is only recommended to use the real gas model in the jet utility program for hydrogen gas. It is possible to set the co-density via the generic gas type (see jet-in-generic above), but please use this functionality with care. It is recommended to check the phase-diagram (pressure-temperature-phase) to ensure that the fluid is in the gas phase both in the reservoir and at the nozzle throat.

The pressure and temperature (and density) in the reservoir and at the nozzle throat (sonic condition, i.e. Mach number M=1) are printed in the heading of the output from the jet utility program (see extract below).

```
...# GENERAL:
#...# RESERVOIR:
#...# MACH:
...# pressure = 800.000 barg
#...# temperature = 20.000 C
#...# density = 43.889 kg/m3
#...# pressure, M=1 = 374.061 barg
#...# temperature, M=1 = -38.041 C
#...# density, M=1 = 29.811 kg/m3
...```

The co-density for the Abel-Noble EOS may be approximated by adapting the compressibility factor (Z) to fit results obtained by either Peng-Robinson EOS or Helmholtz EOS (both are capable of a more accurate representation of the real gas state, but they are not available in the jet utility program).

Below is a short list of approximated co-density for some species:

<table>
<thead>
<tr>
<th>Species</th>
<th>Approximated co-density</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO2</td>
<td>1899.93</td>
</tr>
<tr>
<td>O2</td>
<td>1891.56</td>
</tr>
<tr>
<td>N2</td>
<td>1300.85</td>
</tr>
<tr>
<td>AIR</td>
<td>1391.91</td>
</tr>
<tr>
<td>C3H8</td>
<td>829.22</td>
</tr>
<tr>
<td>CH4</td>
<td>670.56</td>
</tr>
</tbody>
</table>

6.2.1.1 Grid refinement for time-dependent leaks

When employing the jet utility program to model a time-dependent leak, the area of the jet, AREA (m²), varies as function of time. If the area of the Control Volume (CV) face at the jet axis (the CV face normal to the jet axis) is smaller than the maximum area of the jet leak (in general the maximum value of AREA (m²) is the initial value), then the Flacs simulator will warn about LEAK EXCESS AREA. You can avoid this by defining the grid such that the CV face at the jet axis is slightly larger than the initial (maximum) area of the jet leak. This is expected to give an optimal grid resolution around the jet axis during the whole release period of the time-dependent leak.
6.2 Release source modelling

6.2.2 Jet2

The FLACS-CFD package includes a second, newer version of the jet utility program utility, called jet2. This new tool has been developed mainly for implementing software-architectural changes that will make the utility easier to extend in the future. As of FLACS v10.7, the differences between jet and jet2 are minimal. This new utility is used by CASD and RISK applications and is also recommended for users.

The new utility can be executed using the command

Linux:

> run jet2 < jet-in > jet-out

Windows:

> jet2 < jet-in > jet-out

Input files for jet2 are slightly different from the input files for jet. Here is an example:

```
y ! define initial mass flow rate (y) or nozzle diameter (n) (NB! case sensitive)
cl-file ! output format (-, *, cl-file, ...)
METHANE=0.8,HENDECANE=0.15,TOXIC TETRAETHYL LEAD=0.05 ! gas type (AIR, METHANE, ...)
100 ! reservoir volume (m3)
50 ! reservoir pressure (barg)
30 ! reservoir temperature (C)
1 ! atmospheric pressure (bara)
30 ! atmospheric temperature (C)
0 0 ! heat transfer coefficients (J/s) and (J/s/K) (J/s/K)
10 ! initial mass flow rate (kg/s) OR nozzle diameter (m)
0.85 ! discharge coefficient (-)
0 ! start time (s)
1 ! time step (s)
10 ! number of iterations (-)
'J+X' ! leak control string
1e-5 ! shutoff pressure (barg)
1e5 ! release mass (kg)
0.1 ! relative turbulence intensity RTI (-)
*D ! turbulence length scale TLS function
```

Besides the different format, a new feature available in jet2 is the ability to specify a desired initial mass flow rate instead of the nozzle diameter. This feature can be turned on or off by setting the first parameter in the file to either 'y' or 'n', respectively.

An additional capability of jet2 is the ability to add custom gas types. This option can be enabled by specifying the -add option when running jet2. The program will then query the user for the gas name, molecular weight and kappa value.

The jet2 utility can also process input files for the old jet program. This can be achieved by specifying the -compatibility option, i.e.

Linux:

> run jet2 -compatibility < jet-in > jet-out

Windows:

> jet2 -compatibility < jet-in > jet-out

In contrast with the jet utility program, jet2 computes the KAPPA values on the fly, depending on the reservoir temperature. Therefore, jet2 is in general more accurate than jet, especially for very low or very high temperatures.

Table Species supported in the Jet and Flash utilities shows which species are available with the Jet2 utility. All the toxic components (cf. table Predefined toxic substances) are supported as well, but their name must be preceded by the keyword TOXIC, for instance TOXIC ACROLEIN, TOXIC HYDROGEN CYANIDE, etc.
Note:
The jet2 utility program can use two different models to compute the expanded-jet properties:

- the model by Birch,
- the Ewan-Moodie model.

The Birch model is the default, while the Ewan-Moodie model can be selected by specifying the command-line option `-ewan_and_moodie` to both the jet and jet2 programs.

### 6.2.3 Real gas effects modelled in the jet utilities

The jet utilities (Jet, Jet2) assume ideal gas behaviour for all gases by default. In addition, an advanced option is available to include real gas effects for pure hydrogen, but this option must be explicitly enabled by the user, either under advanced options in the leak wizard GUI or with an additional key when running the jet utilities from the command line.

The flow is assumed to be reversible/isentropic up to the shock, so the Joule-Thomson effect is not present. If the reversible/isentropic flow assumption is compatible with the real flow situation (perfect nozzle: no 'throttling/friction' losses) then there should be no Joule-Thomson effect.

If the real flow situation is through a valve (throttling device) then the Joule-Thomson effect will be present. This would be a flow with constant enthalpy (assuming no heat exchange with the surroundings) and with non constant entropy - this situation is not modelled in the jet utilities.

The real gas "equation of state" (EOS) employed in the jet utilities is the so-called Abel-Noble EOS.

\[
p = \rho R T : \text{ideal gas EOS}
p = \rho R T Z, \quad Z = 1 + p/(\text{co-density} R T) : \text{real gas EOS (Abel-Noble)}
\]

Where:

- \( p \) = pressure (Pa)
- \( \rho \) = density (kg/m³)
- \( T \) = temperature (K)
- \( W \) = molar mass (kg/kmol)
- \( R_{\text{uni}} \) = universal gas constant (J/(K.kmol))
- \( R = \text{specific gas constant}, \quad R = R_{\text{uni}}/W, \quad (J/(kg.K)) \)
- \( \text{co-density} \) = the co-density (kg/m³) in the Abel-Noble compressibility factor
- \( Z = \text{Abel-Noble compressibility factor (-)}, \quad Z = 1 + p/(\text{co-density} R T) \)

\( R_{\text{uni}} = 8.314.472 \ (J/(K.kmol)) \)
\( W \) for H₂ is 2.01594 (kg/m³)
\( \text{co-density} \) for H₂ is 130.03 (kg/m³)

The plot below shows a comparison between ideal gas and real gas density as function of pressure for hydrogen.
Figure 6.3: Jet utilities: ideal gas and real gas density as function of pressure for hydrogen

The plot below shows a comparison of ideal gas and real gas mass flow rate for hydrogen gas released from a 0.1 m³ reservoir through a 4 mm diameter nozzle. The initial conditions in the reservoir are 800 or 200 barg and 20 degrees C.
A detailed description of the implementation of Abel-Noble EOS in the jet utilities can be found in the PhD thesis of Prankul Middha (University of Bergen, Norway, 2010).

An extensive description of the Jet utilities (ideal gas model) can be found in the Technical Reference chapter.

### 6.2.4 Entrainment modelling for leaks

The **entainment model** in FLACS-CFD is designed to reduce the computation time of dispersion simulations involving high speed (i.e. \( > 100 \text{ m/s} \)) leaks, while preserving a reliable description of the flammable cloud. The entrainment model allows to generate the leak conditions at a downstream position where the jet diameter has increased and its velocity has decreased. Comparing to a dispersion simulation performed with the standard jet leak definition a typical speedup factor between 2 and 6 can be achieved for scenarios where the leak speed is the limiting factor for the time step, according to the CFLV condition. The grid refinement around jet leaks (cf. default cell size) can also be relaxed, making this approach particularly suitable when the dimension of the release area is significantly smaller than the grid-cell size.

#### 6.2.4.1 Entrainment functionality

The entrainment functionality is a modelling technique for high speed jets based on the jet integral model. The jet equations are integrated from the conditions at the leak position, specified in the leak file, to the downwind position where the jet cross-section has expanded to the grid cell size. The spatial domain where the entrainment model is applied, compared to the leak and grid-cell sizes, is shown graphically in the figure below.
6.2 Release source modelling

The entrainment model is coupled with the Flacs solver in two ways:

- the jet conditions \( J_2 \) after entrainment of the surrounding fluid, at position \( E2 \) in the figure above, define a source term in Flacs,
- the gas composition and temperature in the jet surroundings, as computed by Flacs, define the conditions of the entrained fluid in the entrainment model. In Flacs, the entrainment inflow is included as sink term \( S \), subtracting from the surrounding flow the same amount of mass that is absorbed by the jet.

In accordance with the free flow assumption made in the entrainment model, it is required that the trajectory from \( E1 \) to \( E2 \) in the figure above does not interfere with obstacles. You have to ensure that this condition is fulfilled, see the section on grid guidelines for modelled-entrainment leaks. The location \( E2 \) is calculated based on the maximum mass rate conditions specified in the leak file. The jet state after entrainment is based on the outflow conditions at the specific time as defined in the leak file.

Note:

For releases of substances stored under pressure, the entrainment model does not compute the expansion from the nozzle conditions to ambient pressure. The jet utility program can be used, as usual, to define the 'expanded jet' conditions and generate the leak file.

Some particular types of source term definitions, specified by the LEAK_CONTROL_STRING, as described in sections leak file and area leaks, are not compatible with modelled-entrainment leaks. Those are:

- fan-type leaks
- suctions
- area leaks

The entrainment functionality is applicable only to axis-aligned leaks and not to oblique jets and is currently not available for Flacs inert.

When considering replacing a standard jet leak with a modelled-entrainment leak, you should first take into account that the higher the leak velocity is, the higher will the expected simulation speedup be. In general, it is not worth to use the entrainment model when the leak velocity is less than 100 m/s. If a gas release from pressurised conditions is considered, this condition is always fulfilled.

If the leak size is such that the grid refinement around the leak would lead to a significant reduction of the local cell size, with a consequent increase of the computation time, you can opt for the modelled-entrainment description if a free development region of at least 22 diameters is available in the leak downstream field. Refer to paragraph choosing between jet and modelled-entrainment leaks for a description of the entrainment functionality applicability.

The modelled-entrainment dispersion is consistent with the standard jet leak dispersion in terms of flammable cloud extension and spatial distribution. Nevertheless, it has to be taken into account that, due to the coarser
grid, the resulting jet centreline concentration in the near-field is diluted when the entrainment model is applied. For this reason, it is not recommended to use the modelled-entrainment option when the objective of a study is to analyse the near-field concentrations.

6.2.5 Grid guidelines for modelled-entrainment leaks

6.2.5.1 Entrainment model

When the entrainment functionality is activated, the entrainment model is used to integrate the jet conditions to the final state in which the jet cross-section equals the outflow grid face area. The standard recommendations for the grid can be relaxed to allow slightly coarser resolution around the leak. However, special grid guidelines are required to guarantee the necessary distance between the new jet position and the nearest obstacles and to limit the application of the entrainment model to the jet-dominated region, where:

- trajectory deflection due to buoyancy and ambient wind is negligible so that the jet direction is unchanged;
- the ambient co-flow or cross-flow velocity is less than the jet velocity, and the buoyancy flux is low compared to the momentum flux so that the local entrainment coefficient (identified by the term $E$ in the entrainment model equations) is approximately proportional to the jet velocity (Rodi, 1982).

Of these conditions, the ambient velocity limit is generally the most restrictive, as the cross-jet entrainment term associated with it is significant compared to the along-jet entrainment term, away from the high-speed region of the jet. By comparing the formulation of the cross-jet and along-jet entrainment terms and taking into account the jet velocity decay, it is possible to derive the jet trajectory position where the magnitude of the two becomes equal. The jet cross-section area at this position and, consequently, the maximum grid-to-leak-area ratio $R$, depends on the initial leak velocity, the wind velocity and the ratio between the leak density and ambient density.

Considering releases from pressurised conditions, which have a typical velocity of more than 100 m/s, the relation can be simplified, combining the effect of speed and density in a single numerical constant and defining a more practical condition for $R$, which depends only on the wind speed:

- light wind, < 4 m/s: no restriction,
- moderate wind, 4 m/s to 8 m/s: 12 < $R$ < 3,
- strong wind, > 8 m/s: entrainment functionality not applicable.

For moderate wind speeds, the expression for $R$ can be written as $R \leq 200/v_w^2$. The local wind speed therefore determines whether or not it is appropriate to implement the entrainment model, and, if it is implemented, which grid refinement factor is appropriate. The local wind speed can be obtained from a previously run ventilation simulation or can be estimated from the boundary conditions, taking into account that obstacles may displace flow and the wind speed is therefore likely to be reduced in obstructed regions.

In scenarios with light wind and large releases of very light or very dense gases, the buoyancy becomes the limiting factor for the grid-cell size. In such cases, for velocities greater than 100 m/s, a factor $R \leq 9$ is sufficient to restrict the application of the entrainment model to the jet region, where the buoyancy effect is negligible.

In all wind and release conditions it is suggested to keep $R \leq 12$ to ensure that the fuel volume fraction at the new jet conditions computed with the entrainment model is large enough to obtain a correct description of the flammable cloud. When the leak definition includes a lean fraction (cf. the .mix option), the flammable volume might be located in the very near field, and a smaller value of $R$ might be required. It is suggested to use the entrainment functionality with care in this particular case.

For correct discretisation of the flow gradients along the jet, the aspect ratio of the refined leak cells must not be too large. In the modelled-entrainment leak case, the leak term is also repositioned along the jet axis, according to the downstream position computed by the entrainment model (refer to section FLACS-CFD v22.2 User’s Manual)
6.2 Release source modelling

Entrainment functionality), using a nearest-cell criterion. It is therefore necessary to refine the grid around the leak position by adjusting cell lengths in the leak direction such that the aspect ratio does not exceed 3. The entrainment functionality is not recommended for large leaks with a diameter greater than 0.65 times the grid spacing in the core domain, since it would imply enlargement of grid cells in the core simulation domain (grid de-refinement). For these cases, a standard jet simulation is recommended.

6.2.5.2 Distance to obstacles

The modelled-entrainment jet is represented on a grid coarser than the leak size, with flattened concentration and velocity profiles in the near-field. Only further downstream the jet profile regains the typical Gaussian shape. Consequently, it is necessary to guarantee a certain distance of the leak from obstructions or partially porous areas: the higher the ratio \( \frac{S_g}{S_l} \) between grid and leak size is, the larger is the required distance from the initial leak position to the closest obstructions in the leak direction. It has been found that the distance 10\( S_g \) is sufficient to obtain a good representation of the jet profile, and thus a good representation of the jet interaction with impinging walls or porous areas.

Before running the simulation, you have to make sure that the required free region 10\( S_g \) is available downstream of the leak, otherwise at runtime the warning message

```
*** (LEAKS) LEAK % OBSTACLES TOO CLOSE TO LEAK ORIGIN
```

will be displayed. Similarly, if the region between the initial leak position and the new location computed by the entrainment model is (partially) blocked, the error message

```
*** (LEAKS) LEAK \% PARTIALLY BLOCKED CELL ...
```

is output. In this case, since the free-flow assumption of the entrainment model is invalidated, the simulation stops.

6.2.6 Choosing between jet and modelled-entrainment leaks

This section describes the procedure that you should follow to decide whether a leak can be modelled with the entrainment functionality or not. The indications provided here have to be integrated with the grid and application guidelines provided in the sections Entrainment modelling for leaks and Grid guidelines for modelled-entrainment leaks.

Use the modelled-entrainment leak when all the following conditions are met:

- the target of the dispersion simulation is the definition of the flammable cloud extension and spatial distribution (conversely, for accurate concentration results in the near field, e.g. comparison of the model with experimental data, use the standard jet leak type);

- the leak is not situated in a highly congested area and a free development region (at least 22 leak diameters, corresponding to 10 times the grid-cell size when the grid to leak area factor is 3) is available before the leak flow interacts with obstacles. In highly congested areas, the leak size has to be small in order to fulfill the free development requirement;

- the leak velocity is more than 100 m/s;

- the leak has a medium or small area (< 0.65 times the base grid size);

- the local wind speed at the leak location is less than 8 m/s.

In all other cases use the jet leak type.
6.2.6.1 Best practice

When defining modelled-entrainment leaks:

- use the jet utility program to compute the “expanded” leak conditions;
- check that the conditions for using the entrainment functionality are met (refer to the previous paragraph);
- refine the grid using a default grid area to leak area ratio of 9 or a smaller value if
  1. grid spacing > obstacle distance / 10;
  2. grid spacing > \(200/v_{wind}^2\);
(cf. Grid guidelines for modelled-entrainment leaks)
- when the setup is completed run the simulation and verify any error or warning message in the log file (rt-file).

6.2.6.2 Particular scenarios in which the entrainment functionality is not applicable

Some specific scenarios are not compatible with the entrainment model formulation and implementation. Those cases include:

- a leak spanning more than one grid cell, represented as a set of adjacent leaks, since the entrainment hypothesis is invalid;
- the grid-cell sizes at the leak position in the two cross-directions are not the same, therefore the jet is not axisymmetric;
- the release occurs in regions where strong recirculation brings large volume fractions of the fuel back to the proximity of the release point. This situation might occur in closed or semi-closed spaces, when the released fuel volume is comparable to the room volume. Recirculation is taken into account in the entrainment functionality, but the entrainment flux is described as a single point sink term instead of a flux through the jet boundary. For this reason, when the recirculation effect becomes large and the fuel fraction in the entrained fluid exceeds the 10% Flacs issues the warning message

*** (LEAKS) LARGE FUEL FRACTION IN ENTRAINMENT SINK TERM.

Finally, it is not advised to use the entrainment functionality with hydrogen releases. Indeed, the mass entrainment coefficient defined in the model, according to the formulation by Ricou & Spalding (1961), seems to be unsuitable for the low fluid density and large velocity characterising this case (Xiao et al., 2011)

6.2.7 Flash

The flash program is a utility that computes the physical properties of flashing releases of pressurised liquefied gas. The term flashing is usually used to describe vapour formation by pressure changes. Many materials (such as propane, ammonia or chlorine for example) commonly stored as pressurised liquids in the industry can flash as released into the atmosphere. For such pressurised storage conditions the release has, in the free atmosphere, the appearance of a two-phase...
jet composed of droplets and vapour. The image below summaries the thermodynamic state of the material as the distance with the release location increases.

![Figure 6.6: The Flash utility](image)

The end of the near field region is defined as the position where all the liquid in the jet has changed phase. At this position, denoted \( x_f \), the jet is in a single vapour phase and assuming that all the required properties needed to define a gas leak in FLACS-CFD are known, the flashing release can be treated as a so-called jet leak in FLACS-CFD.

The following command starts the flash program.

Linux:

```bash
> run flash
```

Windows:

```bash
> flash
```

Table **Species supported in the Jet and Flash utilities** shows which species are available with the Flash utility. The following inputs are needed:

- Area of the exit orifice
- Temperature of the liquefied gas at the exit orifice
- Value of the discharge coefficient (by default this value is set to 0.62)
- Mass flow rate or pressure at the orifice
- Temperature of ambient air

The species 'user' is a user-defined (generic) type of gas, where values for 12 extra parameters (in addition to common parameters for pre-defined type of gas) are given:

- Molecular weight [g/mol]
- Normal boiling point [K]
- Critical temperature [K]
- Critical pressure [Pa]
- Density of liquefied gas [kg/m\(^3\)] at normal boiling point
- Specific heat of vapor of released gas at constant pressure [J/(kg K)], at normal boiling point
- Specific heat of liquefied gas at constant pressure [J/(kg K)], at normal boiling point
• Latent heat of evaporation [J/kg] at normal boiling point
• Thermal conductivity of liquefied gas [W/(m K)] at normal boiling point
• Surface tension between liquefied gas and its vapor [N/m] at normal boiling point
• Factor in reaction scheme between fuel and oxygen. For example for propane: \( \text{C}_3\text{H}_8 + \gamma \text{O}_2 \rightarrow 3 \text{CO}_2 + 4 \text{H}_2\text{O} \) with \( \gamma = 5 \). The factor \( \gamma \) is used when calculating the equivalence ratio.
• Dynamic viscosity of vapor of released gas [kg/(m s)], at normal boiling point

If you consider to apply an inert gas (for instance CO2) as a ‘user’ type of released gas with the flash utility: for the factor \( \gamma \), you may specify an arbitrary number, e.g. the value 5, since the output information from the flash utility program about the Equivalence Ratio is neither relevant nor used to prepare scenario setup for a Flacs simulation where the fuel composition is pure CO2 (or other inert gas). Output information about the Equivalence Ratio can just be ignored.

From the inputs the flash program gives the following outputs:

• Position \( x_f \) where all the fluid is in a single vapour phase
• Area of the jet at the position \( x_f \)
• Velocity of the jet at the position \( x_f \) and mass flow rate
• Mass fraction of air and released material at the position \( x_f \)
• Mass fraction of released material that rained-out and formed a pool on the ground.

In the previous list, the first three outputs can directly be used to define a leak in FLACS-CFD, the fourth output allow deriving the value for the equivalence ratio of the released material in FLACS-CFD and if significant, the last output should be part of a pool setup (see pool model).

An extensive description of the Flash utility can be found in the Technical Reference chapter.
The FLACS-CFD package includes a second, newer version of the Flash utility, called flash2. This new tool has been developed mainly for implementing software-architectural changes that will make the utility easier to extend in the future. This new utility is used in CASD leak wizard for liquid phase calculations. For command line users there is no difference between flash and flash2 utility except the name of executable.

## 6.3 Modifying simulation files

### 6.3.1 cofile (deprecated)
The cofile utility extracts data from the FLACS-CFD geometry file cofile. The program reads a cofile and writes obstacle size distribution to the screen.
The cofile utility counts the number of primitives and calculates the length and surface area of boxes and cylinders. The object surface area per \( m^3 \) is believed to be a better measure of congestion level than \( \text{pipelength/m}^3 \), and may be useful for ACM and RCM approaches, see (Hansen, 2010). Be aware that the areas reported by cofile for cylinders will be the same as for boxes, and there may be a need to scale these with a drag coefficient. When using such approaches to define congestion level, one should make sure to exclude the surface areas of very large objects, e.g. walls and decks.
The specification of the option \( \text{list}=? \) with the cofile utility results in a full list of primitives. cofile can also list the total length of cylinders/boxes for each diameter size classes providing the option(s) \( \text{classes_cyl=c1,c2,.., or/and classes_box=b1,b2,..} \).
The following command lists all the options available for use with cofile.

**Linux:**

```bash
> run cofile
```

**Windows:**

---

FLACS-CFD v22.2 User’s Manual
6.3 Modifying simulation files

> cofile

The output of this command is:

```
FLACS cofile (version 2.1, 2013-04-23)
Copyright 2013, Gexcon AS
usage: cofile file_name [options ...]
options (enter in any order after file_name):
  region=x,X,y,Y,z,Z region of interest
  classes_cyl=c1,c2,.. classes of cylinders
  classes_box=b1,b2,.. classes of boxes
  inch=0.0254 set inch to meter scale
  accu=0.001 set size accuracy
  plot=0/1 plot=no/yes (using gnuplot)
  list=0/1 list=no/yes
  silent=1/2 do not write to screen/file
  overlaps=1/2/3 overlaps=INCLUDED/EXCLUDED/BOTH
  hues=... filter by color hue (e.g. hues=0:60,120:180)
  skip=... skip object type(s) boxes/cylinders/ellipsoids/generals
  mindia=0.001 minimum pipe diameter
  maxdia=1000 maximum pipe diameter
  decimals=1,3 number of decimals in output format
```

Abstract:
The program reads a FLACS-CFD co-file and writes obstacle size distribution:
- full list of primitives when the list=1 option is given
- total length and cross-section areas of cylinders for each diameter size class
- total length and cross-section areas of boxes for each diameter size class

Output is by default written to the screen and to a file_name.info file

Note the following:
- The box diameter is the length of the medium-length side of the box
- The cross-section area is the area of the projection in the X, Y and Z direction
- The counting of the cross-section area is done in each of the 6 directions +/-X, +/-Y and +/-Z

A technical description of cofile can be found in Technical Reference.

6.3.2 cofile2

cofile2 is an improved version of the older cofile utility, and is used by the Statistics option in CASD. The handling of overlapping primitives has been improved, at the cost of increased computation time. cofile2 can be used on both co-files and directly on a CASD database. The following command lists all the options available for use with cofile2.

Linux:

> run cofile2

Windows:

> cofile2

The output of this command is:

```
usage: cofile2 [-h] [-i INCLUDE_REGION] [-e EXCLUDE_REGION] [-f PREFILTER]
              [-o OUTPUT_FILE] [-q] [-t THREADS]
              [settings] filename
```

positionals arguments:
  settings Settings filename.
  filename Filename of co-file or database.

optional arguments:
```
Utility programs in FLACS-CFD

-h, --help show this help message and exit
-i INCLUDE_REGION, --include-region INCLUDE_REGION
Region of interest. In the format (x1,y1,z1,x2,y2,z2).
-e EXCLUDE_REGION, --exclude-region EXCLUDE_REGION
Region to be disregarded. In the format (x1,y1,z1,x2,y2,z2).
-f PREFILTER, --prefilter PREFILTER
Filter primitives before counting.
-o OUTPUT_FILE, --output-file OUTPUT_FILE
Output file for report.
-q, --quiet Don’t print additional info.
-t THREADS, --threads THREADS
Number of threads to use for counting.

Warning:

Please note that when specifying regions for cofile2 the format of the region is ”x1,y1,z1.x2,y2,z2”, and is different than the format used in the deprecated cofile (deprecated) utility.

Any number of include or exclude regions can be defined. The settings argument allows loading of statistics settings defined and saved in CASD.

The functionality is also available in the FLACS-CFD Python API. The cofile2 source code, available in the python/cofile.py file in the FLACS-CFD installation, serves as an example on how to use the API.

6.3.3 comerge

The comerge program is used to create new FLACS-CFD geometry co-files from existing co-files. Consider a job number 999999 with a given geometry. The following command creates a new co-file co888888.dat3 for the job number 888888:

Linux:
> run comerge region=x_min,x_max,y_min,y_max,z_min,z_max co999999.dat3 co888888.dat3

Windows:
> comerge region=x_min,x_max,y_min,y_max,z_min,z_max co999999.dat3 co888888.dat3

Note: To ensure that region=... is a single argument it must either be without spaces or embedded between quotation marks.

> ... region=x_min,x_max,y_min,y_max,z_min,z_max

or

> ... "region= x_min , x_max , y_min , y_max , z_min , z_max"

but not

> ... "region = ...

The geometry of the new job number 888888 is the same than the geometry of the job number 999999 in the region specified in the command line.

Several co-files from existing job numbers can be used to generate a new co-file. The command:

Linux:
> run comerge

Windows:
> comerge

lists all the options available for use with the comerge program.
6.3 Modifying simulation files

FLACS comerge (version 1.3, 2013-04-23)
Copyright 2013, Gexcon AS

usage: comerge [transform] input_file[s] output_file [force]
enter transform before each input_file
force override
region=x,X,y,Y,z,Z region of interest
init identity transform
translate:tx,ty,tz translate in x,y,z directions
turn:axis,angle,x,y,z turn around axis at point x,y,z
axis = x/y/z, angle = +-90*N (+ is CCW)
skip:box skip boxes
skip:cyl skip cylinders
skip:col= skip objects with given colour hue
only:col= only objects with given colour hue
box=cyl convert boxes
beam=cyl convert composite beams (T/I/H/U shaped)
show show the objects to be converted
skip skip the converted objects
max_W=value set maximum beam width
min_L/W=value set minimum beam length/width ratio
min_T/W=value set minimum beam thickness/width ratio (box beams)
max_T/W=value set maximum beam thickness/width ratio (composite beams)

6.3.4 autostop

A script for monitoring ongoing simulations and gracefully stopping them once they have become stationary. It works by monitoring value changes of the user-specified variable over time. The script reads the changes to the variables value over the user-specified time interval and checks if it has remained stationary. Once this condition is satisfied the variable is considered stable and the simulation is stopped.

If run with the --dry_run option autostop can also analyze completed simulations. This is useful when determining if the runtime of a simulation could have been reduced by using the autostop script.

usage: autostop.py [-h] --directory DIRECTORY [--variable VARNAME]
   [--simtype SIMTYPE] [--logfile LOGFILE]
   [--frequency FREQUENCY] [--time_interval TIME_INTERVAL]
   [--minimum_run_time MIN_RUN_TIME] [--monitor MONITOR]
   [--simulate_stop] [--dry_run] [--analytics] [--graphics]
   [--video]

--directory DIRECTORY  The directory where the simulations to watch are.

optional arguments:
  -h, --help             Show this help message and exit
  --variable VARNAME     Variable to use. If not specified uses Q9
  --simtype SIMTYPE      Only stop simulations of this type.
                          Valid types include: ventilation, dispersion,
                          gas explosion, gas explosion (DDT), dust explosion,
                          fire, wind, inert, and blast.
  --logfile LOGFILE      Path to logfile. If no file is specified a message is
                          written to the console instead.
  --frequency FREQUENCY  How long to pause between checking if simulations
                          should be stopped. In milliseconds. If no value is
                          specified frequency is set to 10000.
  --time_interval TIME_INTERVAL
Utility programs in FLACS-CFD

The time interval to use when reading scenario data. In milliseconds. For example, a value of 2000 results in variable data from the last 2 seconds being used to determine if the simulation has reached a steady state. If the current runtime of the simulation is less than this value the simulation is not checked. The minimum recommended value is 30000 (30s).

--minimum_run_time MIN_RUN_TIME
The minimum duration, in milliseconds, a simulation will be run before the scripts starts checking if the simulation should be stopped. If this value is set to 100000(100s) and --time_interval is set to 1000(10s) the earliest time the script will check if the simulation can be stopped will be at time 110s. One use case would be to start checking at the start time of a leak.

--monitor MONITOR
Name of monitor to read variable value from. Uses monitor at index 0 no value is specified.

--simulate_stop
If specified no simulations will be stopped. Instead, a message is written to the log.

--dry_run
If enabled script can be run on a directory containing finished simulations. It will go through all the timesteps starting at --time_interval. When this option is enabled the generation of CC-files is disabled.

--analytics
Only applicable when used in combination with --dry_run. Outputs one line per simulation. Line is space delimited, and has the following columns: jobnumber, keytags, maximum of examined value within the time-window where the simulation would have been stopped (-1 if it would not have been stopped), stop time of simulation (-1 if it would not have been stopped), and the maximum value of target variable for the entire simulation.

--graphics
Only applicable when used in combination with --dry_run. Outputs a plot of the values of the target variable and the time the simulation was stopped.

--video
Only applicable when used in combination with --dry_run. Generates a video showing the decision process of the algorithm.

Example:
> run run_python autostop.py --directory your/dir/Sim --variable FLAM --dry_run --analytics --monitor FUEL --logfile log_autostop

6.4 Post-processing of legacy simulation output

6.4.1 r1file
The r1file program extracts ASCII data from the FLACS-CFD r1-file.
6.4 Post-processing of legacy simulation output

Note:

The `rfile` utility described in this chapter cannot be used to process CGNS files, but the functionality provided by this utility is also provided as part of the FLACS-CFD Python API and should be used instead of this utility to process both `.dat3` and `.cgns` files.

Considering a job number 999999 the basic command is:

**Linux:**

```bash
> run rfile r1999999.dat3 name=NP force
```

**Windows:**

```bash
> rfile r1999999.dat3 name=NP force
```

The previous command creates an ASCII file named `a1999999.NP` containing the time-history of the variable `NP` measured at all the monitor points defined in the job number 999999. The option force overwrites the file `a1999999.NP` if it already exists.

The name of the output file can be set using `"output="`. The following example generates a file `ABC.NP` from a file `r1999999.dat3`:

**Linux:**

```bash
> run rfile r1999999.dat3 name=NP force output=ABC
```

**Windows:**

```bash
> rfile r1999999.dat3 name=NP force output=ABC
```

Typing the command:

**Linux:**

```bash
> run rfile
```

**Windows:**

```bash
> rfile
```

lists all the options available for use with the `rfile` program:

FLACS rfile (version 1.0, March 1998)
Copyright 1998, Christian Michelsen Research AS
usage: rfile file_name [options ...]
options (enter in any order after file_name):
  name=string    variable name
  output=string  output file name
  format=ascii/binary  output format
  force          force overwrite
  time=start,finish  output time range
  monitors=+::,-::,*?,...?  monitor point list:
    += set, add or remove
    :: first:last:step, positive numbers
    * all (same as : or ::)
    , separator
    ? last character, print monitor map
  example "monitors=1:20,-3,5,7,+31,33"
  example "monitors=2:10:2,13,19?"

6.4.2 `r3file`

The `r3file` program extracts ASCII data from the FLACS-CFD `r3` file. It may also process the data in the `r3-file`. 
Note:

The r3file utility described in this chapter cannot be used to process CGNS files, but the functionality provided by this utility is also provided as part of the FLACS-CFD Python API and should be used instead of this utility to process both ".dat3" and ".cgns" files.

Considering a job number 999999 the basic command is:

Linux:

> run r3file r3999999.dat3 name=NP force

Windows:

> r3file r3999999.dat3 name=NP force

The previous command creates an ASCII file named a3999999.NP containing the values of the variable NP over the entire simulation domain defined in the job number 999999. The option force overwrites the file a3999999.NP if it already exists.

When the values of a 3D output variable in FLACS-CFD are written to a file in ASCII format (plain text) for a certain time instance using the r3file utility program, the order of the values for the output variable, for example the overpressure P like in the example command above, is described by the nested loops:

```
DO K = K_min, K_max
  DO J = J_min, J_max
    DO I = I_min, I_max
      WRITE P(I,J,K)
    END DO
  END DO
END DO
```

Thus the inner loop is for I, then a loop for J, and the outer loop for K. Here K correspond to the z-direction, J corresponds to the y-direction, and I corresponds to the x-direction. The values of I_min, I_max, J_min, J_max, K_min, K_max, depend on the considered region of the grid (the region could be the whole computational domain, or a smaller region). For practical reasons the r3file utility program writes (up to) 5 values on the first line, and whenever needed continues on one or more new lines (no more than 5 values on each line).

In addition to the extraction functionalities, the r3file program can process the data of the NFDOSE variable. Assuming a job number 9999999 and that the file r3999999.dat3 contains outputs of the variable NFDOSE at regular time intervals (i.e. the time intervals are given by the DTPLOT variable in the cs9999999.dat3 file) the r3file program can compute an average dose. For example, considering the following command:

Linux:

> run r3file r3999999.dat3 dose=2 name=FDOSE force

Windows:

> r3file r3999999.dat3 dose=2 name=FDOSE force

This will produce an average dose over dose*dtplot seconds which in our case gives a 20 s average period.

Typing the command

Linux:

> run r3file

Windows:

> r3file

lists all the options available for use with the r3file program:

```
FLACS r3file (version 1.5.3, 2017-10-05)
Copyright 2017, Gexcon AS
usage: r3file file_name [options ...]
options (enter in any order after file_name):
  name=string variable name
```
6.4 Post-processing of legacy simulation output

```
output=string output file name
format=... output format
  ascii
  binary
  cloud
  r3file
  monmap
force force overwrite
time=value output at time
interpolate=[0/1] time interpolation
grid=[0/1] grid output
load=n### load rd-file.n###
dump=n### dump rd-file.n###
reset=NU=0,NV=0, reset fields used for dump file
  available for NU,NV,NW,NK,NEPS
dt=value delta time used for dump file
region=x,X,y,Y,z,Z region of interest
resolution=nx,ny,nz resolution of the map
gridfit=[0/1] fit region to grid
dose=Integer steps for dose
dose/time=Integer steps for dose/time
dtplot=value time between plots
dtsteps=value time proximity
time_apor=value apor<value : apor=0
time_vpor=value vpor<value : vpor=0
small_por=value vpor<value : ?por=0
mix=value mixture mole scale
verbose=[0/1] verbose output
line(...) line detector
```

Attention:

In FLACS v10.6r1 and earlier, r3file has a bug that causes wrong data to be exported under certain circumstances: When requesting data for a time greater than the last output time in the r3-file, the r3file utility does not return data from the last time step, but instead from the last but one time step. This will not matter much when output is written at short intervals, but in any case a workaround is to request output at a time just before the last time for which data is written to the r3file (which can, for example, be found in the t-file after # FINAL OUTPUT; leave away the last digit to avoid rounding errors that could trigger the bug). This bug is fixed in FLACS v10.6r2 and later versions.

6.4.2.1 Air change rate

The air change rate (Air Changes per Hour, ACH) is available Flowvis, but it can still be calculated manually using the r3file utility program.

An example of a command using the r3file program to give input when calculating the air change rate is (remember to include VVEC output when running the FLACS-CFD simulator):

```
run r3file r3000001.dat3 region=0,10,-10,6,5,30 name=AVERAGES force
```

Here in this example run no. 000001 is considered. The region in space considered is in x-direction from 0m to 10m, in y-direction from -10m to 6m, and in z-direction from 5m to 30m. The input name=AVERAGES tells the program that you want output containing average velocities and volume fluxes. Results will in this case be written to the output file a300001.AVERAGES. From the output file you can find the open and total volume of your region, and the volume flux entering and leaving your region as function of time (how many time steps depends on how often you have saved 3D field data on the r3-file). Wind flow at normal velocities (but not an explosion) is approximately incompressible flow, so the volume flux entering the region should be equal or nearly equal the volume flux out of the region. Defining the air change rate as the volume flux per unit time entering the region considered divided by the open volume inside this region (or perhaps you will divide by the total volume including also blocked areas inside the region), it should now be easy to calculate based on output from FLACS-CFD.
6.4.2.2 Creating dump file

When running the r3file program using the 'dump=n###' option, a dump file can be created based on data taken from the given r3-file. A requirement is that all the following variables are present in the specified r3-file:

- P = pressure
- RHO = density
- U,V,W = all 3 velocity components
- KIN = turbulent kinetic energy
- EPS = dissipation rate of KIN
- FUEL = fuel mass fraction
- T = temperature

6.4.2.3 Line monitor output using r3file utility

Line monitor output can be obtained either by using a monitor file that is defined before running the simulation, or by using the r3file utility after running the simulation, provided that necessary 3D output data are stored (cf. the note in the section about monitor files). These two possibilities are illustrated by the following examples:

To receive line monitor output directly, the monitor file cs000001.MON must contain, for example, the lines:

```
VERSION 1.0
line(name="Line1",start=1,1,1,end=9,9,9,output="FUEL(ER)")
line(name="Line2",start=1,1,1,end=9,9,9,output="PRESSURE()")
```

If 3D output is specified for the pressure P and fuel mole fraction FMOLE, line monitor output for "Line1", similar to employing the monitor file, can be obtained by giving the command:

```
run r3file r3000001.dat3
  'line(name="Line1",start=1,1,1,end=9,9,9,output="FUEL(ER)")' force
```

The line monitor output will then be stored in the file a3000001.MON. The option for the line monitor has to be enclosed with single quotes. If the option is given without the quotes, then the Linux command interpreter will complain about Badly placed ()'s.

Note that line monitor output obtained from the r3-file contains only the time steps for which 3D output data are stored in the file; this is normally done only at a few times during the simulation, as specified in the scenario file. In contrast, the direct output of line monitor data stores output for all time steps in the file rt000001.MON.

Line monitor output using the r3file utility should be consistent with corresponding line monitor output using a monitor file, but small differences might be seen due to numerical rounding errors that are not exactly the same when running the core simulator flacs compared to the r3file utility. The main advantage using the r3file utility to obtain line monitor output, is that the line monitor(s) can be specified after the simulation run has finished, provided that the necessary 3D output data are stored.

To generate line monitor data for several lines, the existing file a3000001.MON should be renamed to, e.g., a3000001.MON.Line1 (to avoid overwriting the file when the option force is included). Then line monitor output for "Line2" can be obtained with the command

```
run r3file r3000001.dat3
  'line(name="Line2",start=1,1,1,end=9,9,9,output="PRESSURE()")' force
```

which will again store output in the file a3000001.MON.

6.4.3 a1file

The a1file program is used to process ASCII files with multicolumn data, typically the output from the r1file utility.
6.4 Post-processing of legacy simulation output

Note:

The *a1file* utility described in this chapter cannot be used to process CGNS files, but the functionality provided by this utility is also provided as part of the FLACS-CFD Python API and should be used instead of this utility to process both *.dat3* and *.cgns* files.

Consider a job number 999999 and an ASCII file *a1999999.NP* containing the pressure measurements at three different monitor points. The following command writes the time-integrals of the pressure at the three monitor points:

**Linux:**

```
> run a1file a1999999.NP force integrate :1 :2 :3
```

**Windows:**

```
> a1file a1999999.NP force integrate :1 :2 :3
```

The integrals are written in the file *a1999999.NP.out*. Here is another example with the *a1file* program:

**Linux:**

```
> run a1file a1999999.NP force slope=0.25,0.75
```

**Windows:**

```
> a1file a1999999.NP force slope=0.25,0.75
```

The previous example writes the slopes on screen (slope between 0.25 and 0.75 of maximum value). Finally, the last example shows how to process data by combining columns of the data file:

**Linux:**

```
> run a1file a1999999.NP force integrate :a=1-2
```

**Windows:**

```
> a1file a1999999.NP force integrate :a=1-2
```

This last example writes the time integral of the expression ‘*data in column 1 minus data in column 2*’ into the *a1999999.NP.out* file.

The following command:

**Linux:**

```
> run a1file
```

**Windows:**

```
> a1file
```

lists all the options available for use with the *a1file* program:
The a3file utility program can be used to convert from ASCII (text) format to the binary format that Flowvis understands.

**Note:**

The a3file utility described in this chapter cannot be used to process CGNS files, but the functionality provided by this utility is also provided as part of the FLACS-CFD Python API and should be used instead of this utility to process both ".dat3" and ".cgns" files.

Consider the following command:

**Linux:**

```bash
> run a3file
```

**Windows:**

```bash
> a3file
```

gives a short guidance:

FLACS a3file (version 1.2, 2012-09-12)
Copyright 2012, Gexcon AS
usage: a3file file_name [options ...]

file_name: Name pattern a3??????.VariableName
? can be alphanumeric, but must be a 6-digit
job number if porcalc shall be run.
VariableName can be any alphanumeric tag.

options (enter in any order after file_name):
force force overwrite
source= format of the source file
noBoundaryCells (default) data for domain-internal cells only
withBoundaryCells one layer of boundary cells included
verbose[=0/1/2] verbose output

The content of file a3??????.VariableName must be
1) Number of grid cells in the 3 directions (NX NY NZ)
2) NX*NY*NZ numbers listed sequentially, separated by
space or comma:

```
NX NY NZ
  Var(1,1,1),Var(2,1,1),...,Var(NX,1,1),Var(1,2,1),..., 
  ...,Var(NX,NY,1),Var(1,1,2),...,Var(NX,NY,NZ)
```

In the default case (noBoundaryCells), the NX*NY*NZ numbers are
interpreted as all lying within the domain described by the associated
cg-file. If the source option is given the value "withBoundaryCells",
then the values are expected to include one layer of data around the
domain, as is written by r3file if the output region is not restricted.

Output
Files r3??????.dat3 and cs??????.dat3 will be created.

The a3file tool will only overwrite an existing r3-file if the force option is given.
The source option is described with the input format in the following section.
The verbose option (defaults to 0) can be used to obtain some information during the program run, in
particular for trouble-shooting; note that verbose=2 will print out the complete data, as read by a3file.

**Example:** To convert the file a3123456.var, call

**Linux:**

```bash
run a3file a3123456.var
```

**Windows:**

```bash
a3file a3123456.var
```

This will create the files r3123456.dat3 and cs123456.dat3. The grid file cg123456.dat3 with NX*NY*NZ
cells, and optionally the porosity file, must be created before visualising with Flowvis.
6.4.4.1 Input format for the a3file tool

The input file for a3file must have the form $a3?????.VariableName$ where the question marks can be replaced by alphanumeric characters; if other tools are to be used on the products of a3file then it is advisable to replace the question marks with a six-digit job number, as usual. The $VariableName$ is not restricted to FLACS-CFD variables.

The content of input file must fulfil the following template

$$\text{NX NY NZ}$$

$$\text{Var(1,1,1), Var(2,1,1),..., Var(NX,1,1), Var(1,2,1),...,}$$

$$\text{..., Var(NX,NY,1), Var(1,1,2),..., Var(NX,NY,NZ)}$$

The first three numbers are integers ($NX \times NY \times NZ$) specifying the number of grid cells per dimension. Thereafter, $NX \times NY \times NZ$ real numbers have to be listed, separated by space or comma. The ordering of values in the a3file input file has to be the same as described for output of the r3file tool. The file has to contain a newline character at the end of the last data line (i.e. the end-if-file character must be on a separate line than the last data).

The interpretation of the data depends on the value of the $source$ option to the a3file program:

- for $source=\text{noBoundaryCells}$ (which is also the default case, i.e. if no $source$ option is given), the $NX \times NY \times NZ$ values are considered to belong to a domain of $NX \times NY \times NZ$ cells as specified for a FLACS-CFD simulation. The a3file tool will take care of extending the data by one layer of cells around the domain, as required for an r3-file corresponding to the given number of cells.

- if the $source$ option is given the value $withBoundaryCells$, then the values are expected to include one layer of data around the actual domain. This is the format that is written by r3file if the output region is not restricted. Note however, that a3file does not work directly on r3file-produced ASCII files: r3file writes extra textual information to its output files, which is not understood by a3file.

6.4.4.2 3D output file (r3-file) for a user-defined compound variable

It is possible to use the utility programs r3file and a3file to prepare a 3D output file (r3-file) for a user-defined compound variable. This can be done following these steps:

1. Export, using the r3file utility program, data for two (or more) variables. Standard output from the r3file program includes data for the boundary cells of the computational domain. It is in general recommended to ignore data for these boundary cells when considering a user-defined compound variable. This can be done by choosing the region, in all the three axis directions, such that it starts half a Control Volume (CV) length (i.e. half a cell length) from the boundary in the negative axis direction and ends half a CV length from the boundary in the positive axis direction (this will ensure that only output data for internal scalar grid nodes are included in the output from the r3file program).

2. Use your own tool (script, Excel-sheet, etc.) to combine the data for your two (or more) variables into a compound variable that you define. The calculated values of the compound variable should be written to an ASCII (text) file with content conforming to the requirements for input files for the a3file utility program.

3. Use the a3file utility program to convert from ASCII (text) format to the binary format that Flowvis understands.

6.5 CGNS tools

You can use the tools described in this section to manipulate and view the content of CGNS files. The FLACS-CFD package provides the following tools:

- cgnsview - a CGNS file viewer and editor based on Tcl/Tk.
- cgnsdiff - compare 2 CGNS files.
• cgnslist - list the contents of a CGNS file.

• h5clear - can be used to clear lock inside given HDF5 or CGNS file. Can be used as a recovery tool if files are corrupted by an orphaned lock.

• cgnsClear - cgnsClear.bat and cgnsClear.sh scripts - clearing locks using `h5clear -s` command on all CGNS output files for a given job number. Can be used as a recovery tool if files are corrupted by an orphaned lock.

More information about the above tools can be found in A User's Guide to CGNS.

### 6.5.1 Known issues

There is a known issue when running cgnsview on Windows where short paths (also known as 8.3 file names) have been disabled and the installation path of FLACS-CFD contains spaces. This issue can be solved in two ways:

1. Place the installation directory of FLACS-CFD in a path that does not contain spaces.

2. Contact flacs@gexcon.com to get access to a zip file containing the CGNS tools and any required libraries. This file must be extracted to a path that does not contain spaces, and you must run cgnsview from this path.

Closing the simulator, Flowvis or the cgnsview editor without closing the CGNS file, i.e. interrupting or killing the cgnsview process, may result in file getting corrupted and simulation aborted. To recover the file, try using the cgnsClear tool.
Chapter 7

FLACS-CFD Best practice

This chapter presents examples and best practice guidelines for FLACS-CFD. The examples also include simulations that require special variants of FLACS-CFD, such as FLACS-Fire.

7.1 Geometry modelling

7.1.1 Creating openings like vents and doors

There are several ways to create, for example, a door opening in a wall or vessel. This section discusses two different approaches to this problem.

In summary, the main rule is that the grid should fit the vent opening. Then both methods are equivalent. If you have to choose (the vent opening and grid resolution simply do not fit, no matter what you do), then use the approach that will give you a vent opening that is as close as possible to the real one while reducing the number of partially porous cells/faces to a minimum. It is recommended to carry out a small sensitivity study to estimate the effect of the geometry representation.

7.1.1.1 Option 1

Make the wall with three boxes.

The advantage of this method is that there is no left difference operation which might be difficult to create and maintain. The disadvantage is that the opening area may be adjusted by the porosity calculator to match the grid. If the boxes are expanded, then the opening area, which is critical for explosion pressures, may be wrong. Therefore, if this method shall be used, then it is important to make sure that the grid matches the door properly. Alternatively, a pressure relief panel can be used, but again, the opening area must be prescribed properly relative to the grid.

7.1.1.2 Option 2

Define the wall and door as two flat boxes, and subtract the door from the wall with the help of a left difference operation.

This method will give the exact correct vent area (or rather: the correct blockage ratios based on the opening area), irrespective of the grid resolution, since the area of the opening does not get changed due to adjustment to grid lines. There are examples of simulations where high turbulence has been seen outside a vent opening defined in this way. For situations with circular vent openings/panels this method can be used (combined with a wall at least two grid cells thick) to ensure proper representation of the vent opening.

With the second method, you can get sub-grid contributions around the vent opening if it does not hit the grid properly, and subsequently too high turbulence generation and therefore too high pressures.
7.1.2 Left differences closed to the lower boundary

When a left difference is used to define (part of) a flow domain and the boundary of the “hole” made by the left difference object is exactly on the domain boundary, then special care has to be taken to make sure the domain boundary is closed. Otherwise, depending on the boundary condition, flow can occur across a boundary that was meant to be closed.

An example of this might be defining the lower domain boundary exactly on the floor of a cellar which has been defined by a left difference box in the ground. An extra zero-thickness box can be used to block the flow at the domain boundary coinciding with the border of the left difference operation.

7.1.3 Advanced left difference operations

In FLACS v10.8 two significant improvements have been made to left difference operations:

1. Left difference operations can be added anywhere in the primitive list.

2. Left difference operations can consist of any node type, including other left differences.

The first point means that by using the Mark command a left difference can be created in the middle of the primitive list. Prior to this improvement left difference operations could only be created at the end of the primitive list between the last two nodes. An example of this feature can be seen in figure below.

![Figure 7.1: Using the Mark command to create left difference](image)

In this figure the Mark command have been used to mark a cylinder in the middle of the primitive list. The marked cylinder will be used as the solid part when adding a left difference. In addition to marking the cylinder the first box in the primitive list have been selected. The selected box will be used as the minus part when adding a left difference. Adding a left difference at this point will subtract the box from the cylinder and add the left difference operation to the end of the primitive list.

Left difference operations can consist of all node types, including named unions and left differences. In the figure below a nested left difference operation is shown.
7.2 Terrain roughness

In FLACS-CFD, the scenario variable GROUNDROUGHNESS defines the aerodynamic roughness that characterizes the terrain surface throughout the simulation domain. Aerodynamic roughness has a strong effect on velocity and turbulence close to the terrain level. This means that, for releases that occur or develop close to the ground, the resulting extension of the flammable cloud is very sensitive to the ground roughness. Flacscfd adopts a modified form of the wall functions at the terrain surface to account for ground roughness (see Rough wall boundary). To activate the ground-roughness-based wall functions at the terrain surface, a terrain file must be created. This can be done in CASD either by importing a DEM file or by converting a geometric object to terrain (see section Convert). This latter option can be used to define a flat terrain at the bottom of the simulation domain from a box created in CASD’s geometry editor.

7.3 Grid guidelines

7.3.1 General guidance

In FLACS-CFD, the computational mesh is composed of cubic or rectangular grid cells (or control volumes) defined by vertical and horizontal grid lines, i.e., a single-block rectilinear grid. The mesh spacing can be varied in any of the Cartesian directions. However, it is not currently possible to fit the mesh to curved or inclined walls or objects and so these are modelled using stepped walls and/or sub-grid models. Simulations calculated at finer grid resolutions, or over larger domains, take longer to run, see figure below, and so the resolution should be chosen such that a sufficiently accurate prediction is achieved within an acceptable time frame. It is recommended that simulations that are expected to run for days are first calculated for a coarser grid to check the setup and scenario definition (even if the configuration of the
coarser grid does not meet the recommendations outlined in the grid recommendations table).

Figure 7.3: How the computational cost increases with grid resolution for gas explosion simulations from some experiment campaigns in our validation database. Lines are fitted to data from simulations of individual experiments (colours show experiments from the same campaign). All the plotted simulations were calculated using the same Gexcon HPC.

7.3.1.1 Grid domains

FLACS-CFD requires the simulation domain to extend beyond the region(s) of interest to avoid boundary effects impacting on results. For this reason, the grid generally comprises one or more high resolution domains, referred to as core domains, and a so-called stretched domain, which extends from the core domain boundaries to the total domain boundary. Cells in the stretched domain may increase in size as the distance from the core domain edge increases and the total domain boundary is approached.

In addition to the core and stretched domains, some scenarios require an area of local grid refinement, where very high resolution is required to resolve a specific feature, and these areas are referred to as refinement regions.

For pool scenarios, the horizontal and vertical dimensions of the core domain are considered separately and an extra domain, the vertical pool region, is required between the bottom of the pool and the base of the core domain to ensure sufficient resolution for the evaporation calculations. Figures to illustrate the different parts of the grid needed for pool scenarios are provided in Grid domains for pool scenarios.

Large differences in size for neighbouring cells can cause problems for FLACS-CFD, so it is recommended that these are avoided and the Smooth tool in CASD can be helpful for creating a smooth transition between...
7.3.1.2 Geometry considerations for the grid

7.3.1.2.1 Positioning objects
The edges of large objects (objects that occupy more than 1.5 grid cells, e.g., walls and decks) should be aligned with the grid, and ideally positioned so that their edges fall on grid lines. Grid planes for an existing grid can be adjusted using the Add, Position and Smooth options in the Grid information menu in CASD.

If an imported geometry is not aligned with the XY grid axes, the Auto-align geometry can be used to automatically align it.

Porosity is calculated for each grid cell and if the edges of large objects are not positioned on grid lines then undesirable artifacts may result, such as “leaking corners” or enlarged/shrunken vent areas. If a large surface does not sit on a grid plane, then secondary objects may create porosity profiles that lead to inappropriately increased turbulence at the deck surface, e.g., beams may be considered in the calculations to be on top of the deck instead of below it. It is not necessary for small objects to be aligned with the grid unless they are the dominant structures in a scenario, e.g., in a geometry with only a few objects.

For some complex scenarios, it may not be possible to align all large objects with the grid and still comply with the grid recommendations below. In such cases, it is recommended that large objects which are closest to the ignition are aligned, and other objects are aligned as far as is possible. The extent of any misalignment of large objects with the grid should be considered when interpreting simulation results.

7.3.1.2.2 Pipes
The inner diameter of angles and bends should be increased slightly when modelling pipes using a cylinder minus primitives. The solid wall around “minus primitive holes” must have a thickness of at least one full grid cell to ensure no leakage through the wall (two cells may be better when a cylinder is “subtracted” from another cylinder). A check to confirm that the predicted flow behaviour is as expected is recommended.

7.3.1.2.3 Multiple geometry objects
When the geometry is projected onto the grid, it is important to ensure that there are no partial openings.
where two geometry objects (e.g., pipes or vessels) are joined. This should be checked by calculating and verifying the porosities after creating the grid, see the Porosities menu in CASD.

For more information on defining geometry and positioning it appropriately for the grid, see Representing geometry in the Geometry chapter.

7.3.2 Recommended grid configuration

The recommendations in the table below describe the optimum configuration of the grid according to the scenario characteristics. These are based on validation against experiment results and on an analysis of the calculations in FLACS-CFD. For many scenarios, it is not possible to meet all the recommendations, for example orientating the grid so that all leak directions are aligned with grid axes. In such cases, the recommendations should be followed as closely as possible and a grid sensitivity study may be helpful.

Multiple recommendations in the table apply to any individual scenario and the first column can be used to determine whether each specific recommendation is applicable. Some recommendations are based on information that may not be readily available, such as the initial high pressure region for a blast. The notes below the table provide references to sections of this manual where methods for estimating these are provided.

Once configured, some aspects of the grid configuration can be checked using the grid checking option in CASD.

7.3.2.1 Grid sensitivity

Grid sensitivity studies are recommended for dispersion simulations (see below). For explosions, it may be beneficial to perform grid sensitivity simulations if the guidelines result in resolutions that are too fine for practical simulation (e.g., because of memory resources or because cells smaller than 1 cm are not recommended). Our validation for explosions is based on resolving the gas cloud with 15 cells, and simulations generally converge to the experimental results in this region (although they may diverge at finer resolutions for some scenarios). Numerically, the code does not fully converge with increasing resolution, primarily due to interactions between the structured cartesian grid, the geometry, and porosity. Our validation data are used to tune the models in FLACS-CFD so as to achieve our most accurate results when the clouds is resolved with 15 cells. This resolution allows simulations to be calculated within engineering timescales, while being sufficiently refined that small changes in the resolution generally do not result in large changes to the simulated data (for example, moving from 15 cells to 16 or 17 cells). Grid resolutions in the region of that which resolves the cloud with 15 cells therefore constitute a region of local convergence in most cases. In some cases, sensitivity could be looked at over the range of about 14 to 17 cells to determine if more extreme grid sensitivity is seen for a particular scenario.

7.3.2.2 Dispersion scenarios

Gas dispersion is affected by several parameters, including air and gas densities and velocities, and the size, position and complexity of any geometry in the scene. It is important that the grid resolution is sufficient to resolve all expected gradients. Sensitivity to the grid resolution varies between scenarios and so a sensitivity study is generally recommended. However, this is not always practical and several validation studies have
therefore been carried out to determine the recommendations in the table.

7.3.2.3 Using dispersion model output to simulate an explosion

Different grids are appropriate for dispersion and explosion simulations and so results from a dispersion simulation should be dumped and regridded before being used to simulate an explosion (see Combined dispersion and explosion simulations).
## 7.3.2.4 Summary of grid recommendations

These checks ensure that a grid meets the recommendations. They do not check that the grid is as coarse as possible (i.e., a coarser grid may also meet the recommendations).

<table>
<thead>
<tr>
<th>When this applies</th>
<th>Recommendation</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALWAYS</td>
<td>Grid does not extend unnecessarily below an impermeable ground surface (grid and ground cover whole of the horizontal domain)</td>
<td></td>
</tr>
<tr>
<td>ALWAYS</td>
<td>Cell size at all ROJs/1 ≤ 6 m.</td>
<td>This ensures that all values are meaningful (further recommendations below will result in a smaller value for most scenarios). This may need to be increased to avoid an impracticably high number of grid cells for dispersion and fire simulations that cover a very large domain. Cell sizes of up to 30 m have been used successfully for simulations with ROJs/1 that are still in from the source.</td>
</tr>
<tr>
<td>ALWAYS</td>
<td>Sketch factor/1 ≤ 1.2.</td>
<td>Sketch factors/1 beyond 1.2 have not been widely tested and are not generally required. However, it may be necessary to use a slightly higher sketch factor/2 for a dispersion scenario which otherwise requires an impractically number of grid cells, or where the grid has been adjusted to fit a wall. It is not anticipated that results will be adversely affected provided the factor remains below 1.4, but this has not been widely tested.</td>
</tr>
<tr>
<td>ALWAYS</td>
<td>At least 3 cells between the total domain boundary and any ROJ/1.</td>
<td>This makes it less likely that values are impacted by any unphysical boundary effects.</td>
</tr>
<tr>
<td>ALWAYS</td>
<td>Monitor points should not be in the same grid cell as a solid wall (unless the wall is thicker than a single grid cell) or forms the simulation boundary, e.g., the ground.</td>
<td>The wall may stop to a cell wall that reaches the MP is on the other side of it what was intended. Note that MPs may be placed in a cell that is bounded by a wall (or panel) with thickness &gt; 0 if the wall (or panel) is positioned on a grid plane.</td>
</tr>
<tr>
<td>ALWAYS</td>
<td>Monitor points should be in unblocked cells.</td>
<td>Ideally, monitor points should be in porosity/1 cells. If this is not possible, then we should try to minimize the porosity of the MP cell. If neighboring cells have higher porosity, then we should move the MP so that it is just inside a neighboring cell. If neighboring cells also have low porosity, then a finer grid resolution may be needed to capture any space around the MP.</td>
</tr>
<tr>
<td>Explosion, liquid and fire</td>
<td>Cell size in all domains ≥ 0.1 cm.</td>
<td>Cell sizes smaller than this have not been widely tested as there is a possibility that the sub-grid models may fail. If in particular the sub-grid model for premixed combustion (explosion) may adversely overpredict burning velocities for very fine resolutions. Where a scenario is particularly fine-scaled (or where grid refinement is needed around a very small feature, e.g., a small point leak), finer cells may be smaller than this, however it is recommended that results are checked carefully and if any extremes are present that indicate model breakdown, then a coarser resolution should be used. Finer resolutions have been tested for some fire scenarios and accuracy has not been adversely impacted.</td>
</tr>
<tr>
<td>Blast</td>
<td>Monitor points should not be cells with porosity = 0.5.</td>
<td>FLACS Blast treats porosities as binary, so porosity=0.5 cells are ambigious.</td>
</tr>
<tr>
<td>Any scenario with an ignitability/explosive location</td>
<td>Ignition should not be on a grid line and should be in an unblocked cell.</td>
<td>See note above about recommendation for MPs in unblocked cells. For blast scenarios, the explosive location may be in a partially blocked cell but an initial simulation should be used to check that the high temperature and pressure phases do not partially penetrate the wall.</td>
</tr>
<tr>
<td>Any scenario with an ignition</td>
<td>Ignition should not be in the same grid cell as a solid wall.</td>
<td>See note above about the same recommendation for MPs.</td>
</tr>
<tr>
<td>Explosion</td>
<td>Aspex ratio7 for cells in core domain ≥ 2.</td>
<td>The longest cell side should no more than twice the length of the shortest cell side.</td>
</tr>
<tr>
<td>Blast</td>
<td>Aspex ratio7 for cell sides in the horizontal plane in the core domain ≥ 1.</td>
<td>If Z is vertical, then the lengths of the X and Y sides of cells in the core domain should be equal. Cells may be elongated in the Z direction by up to a factor of 2 if the wave propagation in this direction is not of interest (otherwise cells should have an equal length in X, Y and Z).</td>
</tr>
<tr>
<td>Gas explosion with vents/1</td>
<td>Minimum distance from vents/1 to core domain/1 boundary: • 7 cells in vent opening direction • 5 cells in all other directions</td>
<td>Recommended for all dust explosions to ensure that any external explosion is sufficiently resolved. For a gas explosion, an initial coarse-resolution simulation may help to determine whether an external explosion is likely. The possibility of external explosion should always be considered when the test mix includes hydrogen.</td>
</tr>
<tr>
<td>Dust explosion with vents/1</td>
<td>Minimum distance from vents/1 to core domain/1 boundary: • 7 cells in all other directions</td>
<td></td>
</tr>
</tbody>
</table>

---

370 FLACS-CFD Best practice

FLACS-CFD v22.2 User's Manual
### 7.3 Grid guidelines

<table>
<thead>
<tr>
<th>Blast</th>
<th>The initial high-pressure region should be resolved by at least 5 cells in every direction.</th>
</tr>
</thead>
</table>
| Explosion | Core domains** should cover:  
- The expanded gas cloud(s)  
- Any geometry that extends beyond the cloud(s) and is likely to impact on the explosion  
- Any vent***  
- All near-field ROCs** |
|          | Allow for 1% expansion of the gas cloud in every direction in which expansion is possible. |
|          | In some cases with multiple gas clouds, it may be more practical to create separate core domains for the individual gas clouds. |
|          | Some judgement may be required to determine whether geometry beyond the expanded gas cloud is likely to impact on an explosion and a sensitivity study may be helpful. |
|          | See recommendation below for how to deal with far-field ROCs which would result in an impractically big core domain. |
| Explosion | Size for cells covering the expanded gas cloud should be ≤ 0.1% in each dimension, where O is the extent of the expanded gas cloud in that dimension. |
|          | When multiple clouds are included in the same core domain, this recommendation should be met for every cloud. |
|          | The recommended maximum aspect ratio for core domain cells means that clouds with an aspect ratio of greater than 2 should be resolved with more than 15 cells in their maximum dimension. |
|          | In cases with small geometry enclosed within the cloud, a sensitivity study is recommended to ensure that the geometry is sufficiently resolved. |
| Explosion with vents*** (including vents covered by plates) | A minimum of 3 cells is recommended across every vent in every direction. |
|          | When this is impractical, then it is important that vents are covered with at least one cell in every direction that has geometry 1. |
| Explosion with vents*** (including vents covered by plates) | The edges of all vents should be on grid planes. |
|          | This is particularly important if only one cell is used to resolve a vent. |  
|          | When some of a surface constitutes a panel surface that protrudes or is inclined slightly from the surrounding surface, then this may lead to grid planes being incorrectly placed close together. For these cases, it is recommended that the panel surface is adjusted in the geometry to lie in the same plane as the surface that surrounds it. |
| Explosion and blast | Distance between the core domain’s boundary and total domain boundary ≥ 3 – the core domain’s extent in every direction. |
|          | This ensures the likelihood of boundary effects impacting on calculations in the core domain’s and is particularly important for cases with FLACS-FLACS boundary conditions. |
| Explosion and blast with far-field ROCs** | Core domain** cell size should be maintained from the core domain’s to the ROCs**. |
|          | Pressures are calculated at lower resolution over stretched parts of the grid so this recommendation is to avoid grids being damaged. |
|          | If this results in an impractically high number of grid cells, then the grid should be stretched beyond the core domain** until cells have increased in size by a factor of ten. For example, a grid with 1 m cells in the core domain** may be stretched outside the core domain and the cell size modestly, e.g. Cells between the point and any far-field ROCs** should all have a size of 10 m. |
| Dispersion and fire (including post fire) | The leak direction should be aligned with a grid axis. |
|          | The CENO/DGEM setting ensures that this is the case, however it may not be possible for scenarios with multiple leaks. In these cases, direction vectors can be used to specify the direction for leaks that cannot be aligned with the grid. |
| Dispersion and fire | Leak should not be positioned on a grid line in the leak direction. |
|          | For example, a leak that flows in Z direction should not be positioned on a Z grid line. In the simulation, the leak is assigned to a cell and if it is positioned exactly on a boundary then it may be assigned to the adjacent cell and the leak may flow out of the wrong cell. |
| Dispersion and fire with area leaks | Leak edges should be on grid planes. |
|          | When this is not possible, e.g. for an elliptical leak, the leak area should be contained within a whole number of grid cells without necessarily fitting every cell (which cfd will correct for this). |
|          | It is important that there is at least one fully unblocked (i.e., porosity = 1) of over the leak. |
| Dispersion and fire with area leaks | The centre of the core domain** for each leak should be located at the centre of the leak. |
|          | Multiple core domains may be required for scenarios with multiple leaks. |
| Dispersion and fire with area leaks | In the plane of the leak, the core domain** should extend beyond each leak edge by at least one cell. |
|          | All cells in the core domain should be the same size and shape, but this shape need not be cubic since cells may be elongated in the leak direction by up to an aspect ratio of 3. |
| Dispersion and fire with area leaks | Every leak should be resolved by at least 3 cells in the plane of the leak. |
### Dispersion and fire with axis leaks

- Cells in a core domain:айд штейн центр на в ссет овъд овъд овъд в план на докумет." The recommended aspect ratio is a, meaning this also implies a maximum cell size in the dead construction or the core domain.

- Dispersion and fire with point leaks:
  - The cell containing the leak, plus one cell either side of this on the leak plane, plus two cells in front of this (the leak direction) should all be filled, or, cells in this area should all be the same size and shape (do not distort).
  - Cells may be elongated in the leak direction by up to an aspect ratio of 3.

- Dispersion and fire with point leaks:
  - Refilled cells around the leak should have sides of length (0.75 x expanded leak area) / 4 in the plane of the leak.
  - This refilled area is not necessary when only result is in the far-field out of interest.

- Dispersion and fire with point leaks:
  - Point leaks shall not lie in the user grid cell as a solid wall.
  - See note above for the same recommendation for point fires.

- Dispersion and fire with time-varying leaks:
  - User recommendations here should be met throughout the evolution of the leak.
  - For linear profiles for time-varying leaks should be checked.

- Dispersion and fire with time-varying leaks:
  - Leak area(s) should be covered by an odd number of cells in each direction in the plane of the leak.
  - This prevents leaks from sticking to nothing.

- Fire with no pool:
  - Core domain must cover the 3D region occupied by the flame.

- Fire with no pool:
  - Distance from the leak to the core domain boundary in the fire direction = flame length plus 0.8 ft.

- Fire with no pool:
  - Distance from the leak edge to the core domain boundary in the plane of the leak = flame width.
  - For point leaks, this recommendation applies to the distance from the leak centre to the core domain boundary.

- Fire with no pool:
  - Distance from core domain boundary to total domain boundary a extent of core domain.
  - This applies to each direction.

- Fire with no pool:
  - Cell size in the core domain = ± 2 cm.

- Fire with high flow (i.e. including pool fires) and impingement:
  - Any surface or wall that a flame from a pool fire the impingement should be aligned with the grid.

- Pool:
  - Vertical pool region should extend from pool-substrate interface to the height of the basin plus 10 ft.
  - The height of the pool surface should be used if there is no basin.

- Pool:
  - Site of cells in the vertical pool region = a pool diameter + 25.
  - For a spreading pool, the model diameter is the maximum pool diameter.

- Pool:
  - Horizontal core domain should cover the pool surface area.
  - For a spreading pool, use maximum pool surface area.

- Pool:
  - Number of cells to enclose pool diameter ± 15.

- Pool:
  - Cell size in the horizontal core domain = ± 1 ft.

- Pool leak:
  - Distance from pool core to total domain boundary in horizontal plane = ± 5 pool diameters.

- Pool rift:
  - Vertical core domain extends from top of vertical pool region to ± 2 ft flame height.
  - No vertical core domain is generally required for pool scenarios with no fire, provided that the determination for the vertical pool region is met.

- Pool:
  - Total vertical domain extends from the core domain boundary to one (1) the core domain extent for the scenario.
  - For evaporation-dispersion scenarios, substitute vertical refilled region for core domain.
  - The vertical core span is generally required for pool scenarios with no fire, provided that the determination for the vertical pool region is met.

- Geometry present:
  - Large objects should be aligned with the grid and positioned so that edges fall on grid planes.
  - Applies to objects that occupy > 1.5 grid cells, e.g., walls and decks.

- Multiphase geometry objects:
  - Mass-extended partial openings between objects e.g., no leakage.

- Pipes:
  - Solid walls around minus primitive holes must be at least 1 cell thick.
  - This ensures no leakage through the wall (2 cells may be better when a cylinder is subtracted from another cylinder).
Attention:

Pool and dispersion scenarios with non-zero windspeed require a secondary core domain with the same extent and resolution as the primary core domain described in the table. The secondary core domain should be located with its centre offset from the primary core domain centre by a distance equal to the pool radius, in the wind direction. For an area leak, the offset distance should be the distance from the leak centre to the leak edge in the wind direction. This is not needed for jet fire scenarios since the flame shape estimate that is used to set the core domain accounts for windspeed.

Some of the recommendations in the table should be adjusted slightly for the special case of modelled-entrainment leaks, as described in the Entrainment functionality section.

7.4 Combined dispersion and explosion simulations with FLACS-CFD

There are at least three different ways to perform a combined dispersion and explosion simulation in FLACS-CFD:

- Run a dispersion simulation where ignition time and position are set before you start the simulation. If the fuel concentration at the ignition time and position is outside the flammable region there will be no explosion. With this approach it is not possible to use the WIND condition because it enforces a fixed velocity, which is not applicable in the explosion (cf. Wind boundary condition).
- Run a dispersion simulation, look at the results and decide where and when to have the ignition, rerun the dispersion simulation with ignition time and position set. Since you have selected a proper ignition time and position there will be an explosion, but you have spent a lot of extra time to rerun the complete
dispersion simulation. Also with this approach it is not possible to use the WIND condition because it enforces a fixed velocity, which is not applicable in the explosion (cf. Wind boundary condition).

- The preferred method is to run a dispersion simulation, create simulation dump files at selected time instants, look at the results and restart the simulation from the dump file with time closest to the desired time of ignition. This gives you the flexibility to select several ignition positions without having to rerun the dispersion simulations. You can monitor the progress of the dispersion and decide to create dump files also after the simulation has been started (use the cc-file). With this approach it is possible to use the WIND condition during the dispersion simulation and to switch it off (change to EULER) for the explosion simulation.

7.5 Simulation example

7.5.1 Initialisation

Create a directory for the example and copy the files from

```
/usr/local/Gexcon/FLACS-CFD_22.2/doc/examples/ex04_dispersion/*00001*
```

as described in section Creating the work directory. Start the FLACS-CFD RunManager.

7.5.2 Wind and Dispersion Simulations

In the RunManager, use Add Directory to find the directory that contains the geometry files. Use RunManager → Tools → CASD (or click the FLACS-CFD pre-processor icon). Open the file 200001.caj (and ignore any error messages that appear). The geometry is a representation of a full-scale process module. The dimensions are 28 m × 12 m × 8 m.

7.5.2.1 Scenario

7.5.2.1.1 Monitor points and output variables

Define a regular pattern of 16 monitor points inside module (X=3, 9, 15, 21, Y=2, 6 and Z=2, 6). This can quickly be done by using the 'Arrange in grid' functionality found on the context menu which is opened by right clicking with the mouse in the monitor points window.

Measure FMOLE and UVW at monitor points by selecting all monitor points, right click and select 'Edit'. Measure FMOLE, ER and VVEC for 3D-output (see the Single field 3D output section). Remember to hold the CTRL key while selecting multiple variables for output.

![Figure 7.5: Specification of monitor points](image)
7.5 Simulation example

7.5.2.1.2 Simulation and output control  Set \( NPLOT=-1 \), and \( DT\) \( PLOT=2.5 \). The CFLC number can be increased due to grid refinement, see the discussion below and section Simulation and output control for additional details. With refinement factor equal 5 and multiplying with default CFLC value 10 for dispersion simulations, set CFLC=50 (running this specific example simulation, CFLC=100 was used to speed up calculations even more). Set default value for dispersion simulations CFLV=1 (running this specific example simulation, CFLV=2 was used to speed up calculations even more). Set the total simulation time to be Maximum time=75 (see Maximum time).

7.5.2.1.3 Boundary Conditions Define WIND inflow from XLO and YLO, 2 m/s diagonally (use Wind Direction (1,1,0)), use Wind buildup time = 0, NOZZLE at other boundaries. The Wind wizard can be used to set the WIND boundary condition quickly.

![Figure 7.6: Specification of the WIND boundary condition](image)

7.5.2.1.4 Initial conditions Set the initial turbulence low for stability (\( \text{CHARACTERISTIC VELOCITY}=0.1 \), \( \text{RELATIVE TURBULENCE INTENSITY}=0.1 \), \( \text{TURBULENCE LENGTH SCALE}=0.01 \)). For the logarithmic wind profile, set the Reference height to 10, and Ground roughness to 0.01. The Pasquill class must be F. Leave the other parameters unchanged.

![Figure 7.7: Specification of initial conditions.](image)

7.5.2.1.5 Gas composition Specify natural gas (91% Methane, 7% Ethane and 2% Propane) and set \( \text{ER0} = 1e30 \) (pure gas release). The fuel region size should be zero (no gas cloud initially).

7.5.2.1.6 Leak In the leak menu, specify a single leak at the position X=6m, Y=5.05m, Z=2.38m and with the direction +X (use Open sides). Start the leak at \( T=10 \) seconds, for a duration of 40 seconds. The start time is chosen so that the wind field can reach steady state.
Double-click on Outlet. Use a mass release rate of 4 kg/s through a 0.02 m² leak area. Set the initial turbulence for the leak: relative turbulence intensity = 0.2, turbulence length scale = 10% of leak diameter = 0.014, Temperature = 20°C, leave the direction cosines as (0, 0, 0).

7.5.2.1.7 Ignition Specify an ignition time of 9999 seconds (an arbitrary high value after the end of the simulation so that FLACS-CFD does not try to ignite the gas cloud). Leave the ignition position unspecified.

7.5.2.1.8 Gas monitor region Define the gas monitor region to cover the module: position (0,0,0) m and size (28, 12, 8) m.

7.5.2.2 Grid

7.5.2.2.1 Quick grid: simulation volume and grid resolution Start the Quick grid tool from the Grid menu. Set the core domain to coincide with the module, i.e., minimum (0,0,0) m and maximum (28,12,8) m. Set the cell size to 1.333 m. Since the grid is uniform, this will lead to 21 × 9 × 6 grid cells in the core domain. Let the stretched domain start at (-32,-28,0) m and end at (60,40,32) m. Leave the maximum stretch factor at 1.2.

7.5.2.2.2 Grid refinement The grid needs to be refined around the leak to avoid strong dilution, see grid recommendations. This can be done using the automatic grid refinement tool in CASD, or can be done manually following the steps below. The expanded leak area is 0.02 m². The grid cell containing the leak (the 'leak cell'), and one cell either side of it, should each have an area of \((1.25 \times 0.02)^{0.5} = 0.1581 m^2\) in the plane of the leak. The leak cell may be elongated in the leak direction, provided that it's length in this direction is not more than 5 times it's length in the other directions. So, in the leak direction, the cell length should be between 0.158 m and 0.790 m (automatic grid refinement will set it to 0.158 m). This should be applied to the cell that contains the leak, and also to two cells in front of it. Cells in the 3 × 3 block of cells at the leak are now much smaller than their neighbouring cells, so the grid spacing should be smoothly increased until it matches the spacing for the surrounding grid, as shown below:

![Figure 7.8: Sketch of the grid refinement.](image)

7.5.2.2.3 Manually refine the grid

Creating the refinement region

The leak is at Y = 5.05 m and grid lines should be created \((3/2) \times 0.1581 = 0.237 m\) from this in both directions, at 4.813 and 5.287 m. In the Grid menu in CASD, set the grid direction to Y and use Add to add these two grid lines. Select these two grid lines using CTRL and arrow keys (a message in the yellow box below the CASD geometry window shows which grid lines are selected).

Smoothing from the refinement region

Select Region from the Grid menu and enter 3 to create three grid cells of size 0.1581 m. Now smooth the grid between the two significantly different cell sizes. Select grid lines between -1.333 m and 4.975 m so that grid cells a distance of 4-5 cells from the refined cells, and one refined grid cell, are selected. This defines the region over which the grid will be smoothed (the size of the cell at each end of the smoothing region will remain unchanged by the smoothing, see Smooth). The grid spacing should not change by more than 20% from one grid cell to the next (see grid recommendations); if this cannot be achieved within the
selected smoothing region, then the region must be extended further from the refinement region, i.e., to \( Y \leq -1.333 \text{ m} \). The module edges must remain positioned on grid planes, so grid lines are required at \( Y = 0 \) and 12 m. If the smoothing has caused these lines to be moved, then grid planes in the smoothed region should be manually adjusted using **Move** from the **Grid menu**. Repeat these smoothing steps to smooth between the maximum extent of the refinement region and the ambient grid in the positive \( Y \) direction. **All dimensions**

The steps described above for refinement in the \( Y \)-direction should also be followed to refine the grid in the \( X \)- and \( Z \)-directions. No smoothing is required between the refinement region and the ambient grid in the negative \( Z \)-direction because the refinement region in this direction is bounded by the ground. Smoothing should be carried out following the steps described above in the \( X \)-direction, and in the positive \( Z \)-direction. Following all refinement and smoothing, **Information** can be selected from the **Grid menu** to check that the maximum size difference between neighbouring cells does not exceed 20%.

---

### 7.5.2.3 Runtime simulation control file (cc-file)

The last step before starting the simulation is to make a **cc-file**. In the FLACS-CFD RunManager, click on the dispersion job, click parameters, and edit cc-file and type the following (use capital letters and an extra line shift at the end of the cc-file):

```plaintext
NDUMP 1
TDUMP 40
NDUMP 2
TDUMP 55
```

This gives 2 dumps at 40 and 55 seconds that can be used to restart the calculations.
7.5.2.4 Simulation

Start the simulation by clicking on the job and clicking simulate in the FLACS-CFD RunManager.

7.5.2.5 Results

The most important result from this simulation is the gas cloud distribution. It can be studied in Flowvis. Cut plane plots of the gas cloud at times 40 s and 55 s are shown below. The concentrations are plotted in the flammable range for natural gas (i.e. between 5 % and 15 % natural gas).
7.5 Simulation example

Figure 7.11: Gas cloud distribution in the flammable range at time 40 seconds.

Figure 7.12: Gas cloud distribution in the flammable range at time 55 seconds.

7.5.3 Explosion simulations

In explosion simulations, one can compute the effect of igniting the realistic gas cloud that was calculated in a dispersion simulation as described above. To this end, copy the FLACS-CFD input files to a new job number (or use CASD to save the job with a new number). Dump the dispersion results at two discrete times: 40 s and 55 s. The gas cloud at 40 s is used to start an explosion simulation.

7.5.3.1 Scenario and Grid

Use Flowvis to find a suitable ignition position e.g., make a contour plot (2D cut plane) of FMOLE or ER in the leak plane (Z = 2.4 m) to find regions where the concentration is close to stoichiometric (this is expected to lead to the worst-case explosion pressure). In this case, the figure above suggests an ignition position of (23, 4.5, 2.4) m. Certain changes need to be made to the scenario and the grid to make them appropriate for an explosion simulation. The following steps should be followed:

1. Change the grid to create a core domain with uniform cells that covers the module and the flammable part of the gas cloud. Follow the grid recommendations to set the size of cells in the core domain based
on the flammable gas cloud volume that is output from the dispersion simulation. Cells between the core domain boundary and the edge of the total simulation domain should be stretched by a factor of 1.2. Note that the total simulation domain may have to be extended to meet the grid recommendations. If the scenario is opened in CASD, then these grid changes can be made using Quick grid from the Grid menu.

2. Simulation and Output Control: Change Maximum time to $-1$, NPLOT=50, CFLC=5, CFLV=0.5. Change DTPLT back to $-1$.

3. Change output variables (both scalar time and 3D): Use P and PROD (remove the other variables).

4. Change ignition position (see above) and ignition time (40.05 s).

5. Save and calculate porosities.

After following the steps above, click on the job in the RunManager (if it is not visible, click on rescan directory). Click Parameters and define a cc-file. The cc-file should contain only one line (remember to include an extra line at the end):

```
NLOAD 1
```

The last step before running the explosion scenario is to generate a new rd-file for starting the calculation based on the dispersion job. This transfers the required information from the dispersion grid (job number XXXXXXX) to the explosion grid (job number ZZZZZZ).

Linux: In a terminal window, type:
```
> run rdfile rdXXXXXX.n001 rdZZZZZZZ.n001
```
(make sure that you are in the correct directory).

Windows: In a command window, type:
```
> rdfile rdXXXXXX.n001 rdZZZZZZZ.n001
```

### 7.5.4 Simulation

Start the simulation by clicking on the job and clicking simulate in the RunManager.

### 7.5.5 Results

Use Flowvis to analyse the results and generate plots. A scalar time plot of pressure at all monitor points is shown below. The maximum pressure of 1.3 barg occurs at $t = 40.34$ s at monitor point 1.

![Figure 7.13: Overpressures at monitor points as a result of the explosion of the dispersed gas cloud at $t=40$ seconds](image)
7.6 Pool spread simulation

7.6.1 Initialisation

It is recommended to start out with an empty directory for storing the files.

**Linux:**
Make a distinct directory (DIRECTORY_NAME) in which you perform the exercise:

```
> mkdir DIRECTORY_NAME
```

Move into this directory:

```
> cd DIRECTORY_NAME
```

Copy the geometry files (notice the space before the ".").

```
> cp /usr/local/Gexcon/FLACS-CFD_22.2/doc/examples/ex02_pool/* .
```

Start up the FLACS-CFD RunManager:

```
> run runmanager
```

**Windows:**

1. Make a distinct directory in which you perform the exercise: Open the file browser navigate to where you want to create the new directory, right-click and choose → New → Folder.

2. Copy files from `C:\Program Files\Gexcon\FLACS-CFD_22.2\doc\examples\ex02_pool`

3. Start the FLACS-CFD RunManager by clicking the desktop icon, or go to Start Menu → AllPrograms → Gexcon → FLACS-CFD 22.2 → FLACS-CFD RunManager.
7.6.2 Case outline

In the RunManager, use Add Directory to find the directory that contains the geometry files. Use RunManager → Tools → CASD (or click the FLACS-CFD pre-processor icon). Open the file 600000.caj (and ignore any error messages that appear). The geometry is a simple model of an LNG terminal. Consider the following case:

• Leakage of LNG with composition 95% Methane, 4% Ethane, and 1% Propane.

• Leakage rate starts at 300 kg/s for 10 seconds, then it decreases linearly to zero during the next 10 seconds. The source of the leak is located on the deck at position x= 28 m, y = 0 m.

• Wind: 3 m/s at 10 meter above sea from south. Neutral atmospheric conditions.

• Results of interest are the size, shape and location of the flammable gas cloud and shape, location and size of the pool.

Remarks:

600000.caj is the case that should be used for learning purposes. Case 600001 contains the same geometry, but it is made ready for simulation.

7.6.3 Grid

First, a Simulation Volume must be defined. The wind comes from the south, aligned with the Y axis, and the undisturbed wind field must be set at a certain distance upstream of the LNG terminal. One is most interested in the gas cloud on the terminal, but also in high concentrations further downstream should be captured. In the X direction, the domain should cover the area the pool can spread and the volume the cloud may fill in the cross-wind direction. In the Z-direction, the simulation volume should include the surface, the LNG terminal and some space above. Note that LNG vapour is heavier than air.

In the current example, a manual grid setup is demonstrated (i.e. without the Quick Grid tool). The following Simulation Volume is used: \( x_{\text{min}} = -50 \), \( x_{\text{max}} = 90 \), \( y_{\text{min}} = -60 \), \( y_{\text{max}} = 120 \), \( z_{\text{min}} = -1 \), \( z_{\text{max}} = 40 \). In the X-direction, initialise a 1 meter grid: choose Grid → Region → 140. Fine grids give more accurate results and it is recommended to use a grid spacing of about 1 meter or less in the X- and Y-direction for pool simulations in the areas where the pool will spread. It may not be easy to forecast where the pool will spread, but some reasoning may be possible when the leak position and obstacles are known. A 0.5 meter grid is recommended in the near-field of the leakage location (\( X = 28 \)). Use CTRL + arrow keys to define the region 22 m – 32 m and set the number of control volumes of the selected region to 20. Smooth the grid. Change to Y-direction (Grid → Direction → Y direction) and generate a 1 m grid (Grid → Region → 180). To get a 0.5 m grid around the leak location, use CTRL + arrow keys to define the region -5 m – 5 m and set the number of control volumes to 20. Stretch the grid towards the boundaries for \( Y > 30 \) m and \( Y < -30 \) m.

Change to Z-direction (Grid → Direction → Z direction) and generate a 1 m grid (Grid → Region → 41) Just above the pool, a 0.25 m grid is desirable. Select the region \( Z = 10 – 13 \) m and increase the number of control volumes from 3 to 12 in this region. Then smooth the grid. Stretch the grid above the geometry (\( Z = 17 \) m) towards the top boundary.

The grid now has about 460000 control volumes and the porosities should be calculated. Remember to save the scenario.

7.6.4 Scenario

To create a pool scenario select 'Simulation type' Pool in CASD.
7.6 Pool spread simulation

7.6.4.1 Single field 3D output

Use MOUSE+LEFT to select POOL, D, FUEL, FMOLE, T, and VVEC. This will give 3D field outputs for the pool depth, that can be used to visualise the pool, FUEL and FMOLE which give the mass fraction and volume fraction (concentration) of the vapour, T, which is the gas temperature, and VVEC, which is the velocity vector.

7.6.4.2 Simulation and output control

Set

- Maximum time=100 (significantly longer than the release period)
- CFLC=20 (increased because of grid refinement)
- CLFV=1 (standard for dispersion calculations)
- DTPLLOT=5.0

7.6.4.3 Boundary conditions

This is a dispersion scenario with wind at 3 m/s from south and neutral atmospheric conditions. Therefore YLO is the inflow boundary and YHI is the outflow boundary. The flow is parallel to the X boundaries and the ZHI boundary and WIND can be used for these boundaries, too.

1. EDIT the XLO boundary and set XLO="WIND", WINDSPEED=3 and WIND_DIRECTION=(0,1,0).
2. COPY the XLO to XHI, YLO, and ZHI.
3. EDIT YHI boundary and set YHI="NOZZLE".

The settings for the boundary and initial conditions for the wind field can alternatively be produced with the wind wizard: in the Scenario Settings click on Run Wizard and choose Wind Wizard. The setup for the current case is as shown in the figure below.
7.6.4.4 Initial conditions

To get a wind field of 3 m/s at 10 meter height and neutral atmospheric boundary conditions, set:

- CHARACTERISTIC VELOCITY=3
- GROUND ROUGHNESS = 0.001 (water, flat surface)
- REFERENCE HEIGHT = 10
- PASQUILL CLASS = "D" (Neutral, turbulence parameters are calculated by FLACS-CFD)

7.6.4.5 Gas composition and volume

1. Edit the volume fraction according to the LNG composition; METHANE=95, ETHANE=4, and PROPANE=1.
2. Set EQUIVALENCE RATIO=(1E+30, 0)

7.6.4.6 Ignition

No ignition is wanted. Set TIME_OF_IGNITION larger than TMAX, for example TIME_OF_IGNITION=9999.
7.6 Pool spread simulation

7.6.4.7 Gas monitor region

Define the gas monitor region to cover the volume above the deck (i.e. X in (-35,35) m, Y in (-25,25) m, and Z in (11,17) m).

7.6.5 Pool setup parameters

If you are a relatively new FLACS-CFD user then it is recommended to use the Pool section of the scenario menu in CASD (set the "Simulation type" under "Scenario Settings" to "Pool", or right-click in the scenario menu to enable the Pool section). The menu, and the settings and keywords used to configure a pool simulation, are described here.

The following setup is used in the example:

![Figure 7.16: Pool setup in the scenario menu (left), pool leak data (right).](image)

The above settings correspond to the following setup of a spreading pool with non-uniform pool temperature in the POOL section in the cs-file. (The POOL section in the cs-file contains more parameters but the scenario menu in CASD only shows a subset of these unless Show advanced is enabled by right-clicking in the view.)

<table>
<thead>
<tr>
<th>Property name</th>
<th>Property ID</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non spreading</td>
<td>STATIC_POOL</td>
<td>0</td>
<td>This scenario models a spreading pool, and not a non spreading pool</td>
</tr>
<tr>
<td>Uncoupled pool fire model</td>
<td>POOL_FIRE_UNCOUPLED</td>
<td>0</td>
<td>Enable to use the uncoupled pool fire model instead of the coupled one</td>
</tr>
<tr>
<td>Start time</td>
<td>START_POOL</td>
<td>5</td>
<td>Pool begins 5 seconds after the simulation starts in order to establish the wind field</td>
</tr>
<tr>
<td>Initial mass</td>
<td>MASS_POOL_0</td>
<td>0</td>
<td>No initial pool mass</td>
</tr>
<tr>
<td>Mass rate</td>
<td>DMDT</td>
<td>300</td>
<td>Pool leakage rate [kg/s]. This value is overridden by the pool leakage file</td>
</tr>
</tbody>
</table>

FLACS-CFD v22.2 User’s Manual
### Property Table

<table>
<thead>
<tr>
<th>Property name</th>
<th>Property ID</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position</td>
<td>POSITION</td>
<td>28 0 20</td>
<td>(X, Y and Z position of the centre of the leakage area. Z position is used to search downwards until solid ground with space above is found, that is ( \text{PORZ}(K) \geq 0.5 ) and ( \text{PORZ}(K+1) &lt; 0.5 ). In this case, Z should be located above the geometry. If a spill on the sea is considered, Z should be located just above the sea surface, see Setting the pool position.</td>
</tr>
<tr>
<td>Inner radius</td>
<td>RAD_I</td>
<td>0</td>
<td>Radius of the leakage area [m]; to get a circular shape for the spill, it is recommended to set ( \text{RAD}_0 \geq 3\Delta x ). In this case ( \text{RAD}_0 = 4\Delta x ).</td>
</tr>
<tr>
<td>Outer radius</td>
<td>RAD_O</td>
<td>2</td>
<td>Radius of the leakage area [m]; to get a circular shape for the spill, it is recommended to set ( \text{RAD}_0 \geq 3\Delta x ). In this case ( \text{RAD}_0 = 4\Delta x ).</td>
</tr>
<tr>
<td>Ground temperature</td>
<td>T_SOIL</td>
<td>293</td>
<td>Ground temperature [K]</td>
</tr>
<tr>
<td>Heat sun</td>
<td>HEAT_SUN</td>
<td>400</td>
<td>Heat from the sun [W/m²]</td>
</tr>
<tr>
<td>Surface roughness</td>
<td>ROUGH_L</td>
<td>0.005</td>
<td></td>
</tr>
<tr>
<td>Ground type</td>
<td>POOL_GROUND</td>
<td>&quot;CONCRETE,WATER[:1 0.5]&quot;</td>
<td>Concrete everywhere except for the Z-region (-1 m – 0.5 m), where there is water</td>
</tr>
</tbody>
</table>

### 7.6.6 Pool leak file

The pool leakage file \texttt{c1000000.POOL} overrides \texttt{DMDT} from the \texttt{cs000000.dat3} file, see pool leakage file. The time in \texttt{c1000000.POOL} is relative to \texttt{START_POOL} in \texttt{cs000000.dat3}. A constant release rate (300 kg/s) for 10 seconds and then a linear decrease in the release rate during the next 10 seconds is defined by:

- `POOL LEAK FILE`
- `TIME [s]` `DMDT [kg/s]`
  - 0.0 300.0
  - 10.0 300.0
  - 20.0 0.0
  - 200.0 0.0

The data for the pool leak file can also be entered in CASD: right click in the scenario menu and enable the Pool leak menu, which allows to enter the rates (right-click in the menu to enter new lines), see the screenshot above.
7.6 Pool spread simulation

7.6.7 Simulations

1. To activate the FLACS-CFD variant with models for pool spread, define RunManager \(\rightarrow\) Parameters \(\rightarrow\) FLACS-CFD version = pool.

2. Select case.

3. Simulate

7.6.8 Results

The most important result is the gas cloud distribution. It is also important to plot the pool shape, pool mass and evaporation rate in order to verify the source term and to ensure that the pool spreads as expected. In the figures below, volume plots are used to visualise the pool and the gas cloud. In addition to the case described above, a release on the sea at the same X and Y position has been considered. Pools are plotted for depths in the range 1 - 20 mm and gas concentrations above LFL/2 are shown. It is recommended to turn off trilinear interpolation (Plot Specification \(\rightarrow\) Trilinear) when plotting POOL_D.

Figure 7.17: Pool spread on deck. Pool (left) and LFL/2 gas cloud (right).
Figure 7.18: Pool spread on the sea. Pool (left) and LFL/2 gas cloud (right).

In the figures below, pool mass, pool area, and evaporation rate per square meter are shown for pool spreading on concrete (deck) and pool spreading on water (sea).

Figure 7.19: Pool evaporation on concrete.

Figure 7.20: Pool evaporation on the sea.

7.6.9 Setting the pool position

The pool model is two-dimensional in space and a crucial point is the detection of the underlying solid surface seen by the pool model. At present, the surface on which the pool can spread is detected by searching downwards from a vertical position, Z, for all X and Y positions. A solid surface is detected if:
7.6 Pool spread simulation

1. \( \text{PORZ}(K) < 0.5 \), and

2. \( \text{PORZ}(K+1) \geq 0.5 \)

This means that if \( Z \) is located in the open, then the first cell with \( \text{PORZ} < 0.5 \) below \( Z \) determines the surface elevation.

It follows from the inequalities above that walls and bunds must be at least \( 0.5\Delta X \) wide in order to be found by the underlying-solid-surface detection algorithm. An example where bunds not are detected is shown in Figure Bunds not detected. Possible solutions are to increase the bund thickness and to refine the grid.

![Figure 7.21: Bunds not detected because PORZ>0.5.](image)

If \( Z \) is given a high value, the pool sees equipment etc. as solid obstacles even though there is a gap underneath the equipment where the pool can spread, see Gap under tanks. The solution is to set a lower value for \( Z \), for instance just above the green surface.

![Figure 7.22: Gap under tanks not detected.](image)

It should be mentioned that if no underlying solid surface is detected by searching downwards, then an upwards search is carried out from \( Z \). The top of tank 1 in Figure Gap under tanks will be detected for a low \( Z \) value.

Difficulties may be encountered when there are several gaps, for instance if there are several decks. In the example in Figure Unwanted depression and Figure No gap, there is no obvious value for \( Z \). If \( Z \) is located above the tanks, the pool will not flow under tank 1. If \( Z \) is just above the surface, an unwanted depression is generated instead of an obstacle at tank

1. The best solution is probably to set a large value for \( Z \), which was done in the best practice example.
Figure 7.23: Unwanted depression at tank 1.

Figure 7.24: No gap between tank 2 and deck.
Warning:

The pool model does not detect pool depth correctly when there are no grid planes beneath the lowest surface in the geometry. The solution to this problem is to extend the grid below the lowest surface on which the pool can occur.

7.7 How to handle a process stream in FLACS-CFD

FLACS-CFD automatically calculates properties for mixtures of species known to it, and mixtures including hydrocarbons from methane up to dodecane can be defined as fuel in FLACS-CFD (see the section on Gas composition and volume).

Attention:

C5 to C10 alkanes have the same laminar burning velocity as butane; C11 and C12 should not be included in a combustible mixture. Cf. the section on potential problems with heavy hydrocarbons.

A simplification of the definition of the combustible mixture can be obtained by replacing C5+ by C4 (butane) in the way it was suggested before C5+ were available in FLACS-CFD, namely by conserving the number of C’s, e.g., 1 mole of C8 becomes 2 moles of butane, since the oxygen consumption (and FLACS-CFD interpolation rules) depends on the number of Cs. This approach underlies the same limitations mentioned above. In any case, one has to assess how much of the heavy hydrocarbons one wants to include, since at some stage they will remain liquid and rain out if the pressure is low (and/or the fraction of lighter components is low).

As far as inerts are concerned, water vapour should have an effect in between what is seen for CO₂ and N₂ as an inert. For CO₂ typically 4% in the final explosive mixture (including air) would reduce pressures by a factor of 2, whereas roughly 8-9% added nitrogen will be required. One could therefore assume that approx. 6% added water vapour in the final gas-air mixture would have a similar effect, reducing explosion pressures by a factor two. The background humidity in air, which can of course vary with temperature and climate, is not considered when doing explosion modelling, i.e. FLACS-CFD is calibrated for a typical relative humidity level (and variations from this may be part of uncertainty in predictions).

Note that nitrogen cannot be chosen as a gas. If one wants to specify N₂ in a fuel mixture, the easiest way will be to redefine the AIR so that your final composition has the correct amount of nitrogen. For dispersion simulations, nitrogen can be replaced with a mixture of other gases (CO or ETHYLENE has the same molecular weight) or a user-defined gas can be defined.

7.8 Fire simulation

7.8.1 Initialisation

It is recommended to start with an empty directory for storing the files.

Linux:
Make a distinct directory (DIRECTORY_NAME) in which you perform the simulation:

> mkdir DIRECTORY_NAME

Move into this directory:

> cd DIRECTORY_NAME

Copy the geometry files (notice the space before the ".").

> cp /usr/local/Gexcon/FLACS-CFD_22.2/doc/examples/ex05_fire/* .

Edit/verify the cs-file with your favourite editor.

> vi cs000004.dat3
Start up the FLACS-CFD RunManager (assuming that you have set up an alias for the \texttt{run} script):

\begin{verbatim}
> run runmanager
\end{verbatim}

\textbf{Windows:}

1. Make a distinct directory in which you perform the exercise: Open the file browser, navigate to where you want to create the new directory, right-click and select \textit{New} \rightarrow \textit{Folder}.

2. Copy the files from

\begin{verbatim}
C:\Program Files\Gexcon\FLACS-CFD_22.2\doc\examples\ex05_fire\*00001*
\end{verbatim}

(00001 means all files containing the text "00001").

3. Start the FLACS-CFD RunManager by clicking the desktop icon, or go to Start Menu \rightarrow All Programs \rightarrow Gexcon \rightarrow FLACS v22.2 \rightarrow FLACS-CFD RunManager.

\section*{7.8.2 Grid}

- First, a simulation volume must be defined, see \textit{Simulation volume}.
- As for other scenarios, refine the grid around leaks; far from leaks a coarse grid is sufficient. It is recommended to follow the \textit{grid recommendations} and to do grid sensitivity studies for fire simulations in the areas close to leaks.
- Calculate the porosities.
- Save the grid specification to the scenario.

\section*{7.8.3 Scenario}

To create a fire scenario select the corresponding simulation type (fire) in the CASD \textit{Scenario menu}. Enable the \textit{write all variables} check-box to get all fire related \textit{single field variables}.

\subsection*{7.8.3.1 Monitor points}

Define the monitor point direction (required to calculate incident radiative flux) along with the position and output variables. The monitor point direction is specified as a vector and determines the heat flux radiometer direction.

\subsection*{7.8.3.2 Single field scalar time output}

Choose the fire-related variables (SOOT, H2O, CO2, CO, VFSOOT, VFH2O, VFCO2, ABSCOF, RADSRC, QRAD, QCONV, Q, QDOSE) required for the analysis to view the results for the relevant monitor points.

\subsection*{7.8.3.3 Single field 3D output}

Left-click to choose TWALL, SOOT, H2O, CO2, CO, VFSOOT, VFH2O, VFCO2, ABSCOF, RADSRC, QRAD, QCONV, Q, QWALL, QDOSE. This will give 3D field output of the variables required for fire simulations.

\subsection*{7.8.3.4 Simulation and output control}

Right-click into this scenario menu and enable \textit{show advanced}. Then apply the following settings:

- CFLC=20 \ (Increase according to grid refinement)
- CLFV=2 \ (Standard for fire calculations)
- Enable the \textit{Wall functions} check-box.
- Enable the \textit{Heat switch} check-box to take heat transfer into account.
7.8 Fire simulation

7.8.3.5 Boundary conditions
The default boundary condition for FLACS-Fire simulations is NOZZLE. The boundary condition WIND can be used to model a specified wind field at selected boundaries.

7.8.3.6 Initial conditions
To obtain better stability, set the initial turbulence to be low:

- \( \text{RELATIVE TURBULENCE INTENSITY} = 0 \) (default value \( k = 10^{-3} \) if RTI=0)
- \( \text{TURBULENCE LENGTH SCALE} = 0 \) (default value \( \varepsilon = 10^{-3} \) if TLS=0)

Leave the other parameters unchanged.

7.8.3.7 Gas composition and volume
- Edit the volume fraction according to the fuel composition.
- Set \( \text{EQUIVALENCE RATIOS} = (1E+30,0) \).

7.8.3.8 Leaks
Always use the leak wizard to specify leaks. There are two ways to represent leaks: point leaks and area leaks.

- In the leak wizard, specify the leak's position (X,Y,Z).
- Enter the leak size (zero for a point leak and non-zero positive values for area leak).
- Specify the leak direction; see Open sides.
- Select the leak type (diffuse, entraining, jet, air, suction, fan, inert).
- Set the leak rate to be manual (known leak conditions), steady state, or time dependent (uses the jet utility program).
- On the next page, select the area leak shape and profile (not applicable for point leaks).
- Depending on the setting for the leak rate, more input may be required.

According to the grid recommendations, the grid should be refined around the leak. Keep in mind that only the grid cell containing the leak and one neighbouring cell on each side needs to be refined and that the grid spacing should be smoothly increased to the prevailing grid spacing. For coupled pool fires, a fine grid is required in the vicinity of the pool surface to anchor the flame, cf. the sections on LNG Pool Fire, Pool spread simulation and Coupled pool fires.

Typical evaporation rates for some of the species in the FLACS-CFD database can be found in the section on Source term modelling for pool fires. For uncoupled pool fires these values may be used to mimic an evaporation pool, by using them as leak rates of an area leak. Another choice is to employ the coupled pool fire model. Here the evaporation rate is calculated by the pool model using a coupling with the radiative heat transfer from the flame to the pool.

7.8.3.9 Ignition
For a fire simulation, you must specify the location and size of the ignition source and also the time and duration for the ignition.

7.8.3.10 Radiation
This sub-menu allows you to select from the available radiation models and specify their parameters (see radiation scenario).
7.8.3.11 Pool fire

In this menu the use of radiative heat transfer to the pool can be chosen and the threshold for the radiative heat transfer can be set. For pool fires in general the FIRESWITCH should be set to 2 under Combustion settings.

FLACS-CFD can model uncoupled and coupled pool fires:

- In an uncoupled pool fire it is assumed that there is no feedback from the pool fire on the evaporation (e.g., burning) rate. It is therefore required to define a uniform evaporation rate as input. In FLACS-CFD, uncoupled pool fires are modelled using either a fixed evaporation rates or basic empirical correlations. Typical evaporation rates for some of the species in the FLACS-CFD database can be found in the section on Source term modelling for pool fires. The uncoupled pool fire model is only suitable for pool fires where the assumption of a uniform fixed evaporation rate holds, i.e. only in relatively open areas.

- In a coupled pool fire, the evaporation (e.g., burning) rate is calculated based on the heat radiation from the fire and from the substrate. This will be more accurate, especially in confined areas or where the evaporation rate is not uniform or unknown. In FLACS-CFD, both the non-spreading and spreading coupled pool fires can be modelled. For the non-spreading pool model, the DMDT setting (i.e. the mass rate inserted uniformly over the initial pool area) is not taken into account and the pool can only burn its initial mass.

Whether a pool is modelled as coupled or uncoupled, it is recommended to use FIRESWITCH=2.

7.8.3.12 Combustion

This sub-menu allows you to select one of the available combustion models. The default and recommended choice is the Eddy Dissipation Concept (EDC). As the EXTINCTION option is still under development it should be deactivated. Set the FIRESWITCH to 2 for both jet and pool fires.

7.8.3.13 Smoke/Soot

This setting allows to select an appropriate soot model. The default and recommended choice is the Formation-Oxidation model.

7.8.3.14 Conduction

The conduction model in FLACS-Fire is under validation. It is recommended not to use the conduction model in this version.

7.8.4 Simulations

To start the FLACS-Fire simulation use the following commands:

Windows:
Start the simulation in the RunManager by marking the job and clicking simulate.

Linux:
Start up the FLACS-Fire simulation (on your local computer):

> /usr/local/Gexcon/FLACS_v22.2/bin/run flacs version fire 000004

If a stack overflow problem occurs see how to Increase the stack size on Linux.

7.8.5 Sample fire simulations

This section presents various test cases for FLACS-Fire.
7.8 Fire simulation

7.8.5.1 SINTEF impinging jet fire

Problem description:

The SINTEF Impinging jet fire (Wighus & Drangsholt, 1993) from high pressure leakages is representative of a severe hazard on offshore platforms and in onshore process plants. The experiments were undertaken to get better knowledge of the effects of an impinging jet fire regarding heat load and the erosive effect from high temperature and a velocity jet.

The propane jet enters horizontally into a box-like target, standing at an angle of 60 or 90 degrees to the jet axis. The dimensions of the box are: width W = 1.5m, height H = 1.5m, and depth D = 0.45m. The jet axis is positioned centrally at 1/2 or 1/4 H, and at distances of L=1.0m, 1.5m, or 2.5m from the wall of the box. A pipe with a diameter of 278.0mm was mounted across the front of the box in three of the tests, at 1/2 or 3/4 of the box height.

Table 7.2: Summary of SINTEF Impinging jet

<table>
<thead>
<tr>
<th>Configurations</th>
<th>SINTEF impinging jet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dimensions of the box</td>
<td>W=1.5 m, H=1.5 m, D=0.45 m</td>
</tr>
<tr>
<td>Fuel</td>
<td>Propane</td>
</tr>
<tr>
<td>Jet velocity</td>
<td>260 m/s</td>
</tr>
<tr>
<td>Jet distance</td>
<td>2.5 m</td>
</tr>
<tr>
<td>Jet Height</td>
<td>0.75 m</td>
</tr>
<tr>
<td>Jet angle</td>
<td>90 degree</td>
</tr>
</tbody>
</table>

Computational details:

In the Figure below the geometry configuration, for the SINTEF impinging jet fire case is shown, including the grid. The number of grid cells used is 57,330. The total number of rays fired is 108.

![Figure 7.25: Calculation domain and grid distribution](image)

Grid:

- Dimensions of the computational domain (m): $X_{min} = -5.0, X_{max} = 5$, $Y_{min} = -5.0, Y_{max} = 5.0$, $Z_{min} = -0.5, Z_{max} = 10.0$
- Total number of cells: 57,330
- Size of grid cells (at the leak location): $200 \times 41 \times 41$ mm
Monitor points:

MONITOR_POINTS
INSERT 1 0.02 0 0.375
DIRECTION 0 0 0
INSERT 2 0.1 0 0.75
DIRECTION 0 0 0
INSERT 3 0.02 0 1.125
DIRECTION 0 0 0
INSERT 4 0.02 0 1.47
DIRECTION 0 0 0
INSERT 5 0.415 0 1.47
DIRECTION 0 0 0
INSERT 6 0.01 -0.1 1.125
DIRECTION 0 0 0
INSERT 7 0.415 -0.1 1.52
DIRECTION 0 0 0
INSERT 8 0.01 0 1.125
DIRECTION 0 0 0
INSERT 9 0.01 0.1 1.125
DIRECTION 0 0 0
INSERT 10 0.9 0 0
DIRECTION 0 0 0
INSERT 11 0.9 0 0.15
DIRECTION 0 0 0
INSERT 12 0.9 0 0.3
DIRECTION 0 0 0
INSERT 13 0.9 0 0.45
DIRECTION 0 0 0
INSERT 14 0.9 0 0.6
DIRECTION 0 0 0
INSERT 15 0.9 0 0.75
DIRECTION 0 0 0
INSERT 16 0.9 0 0.9
DIRECTION 0 0 0
INSERT 17 0.9 0 1.05
DIRECTION 0 0 0
INSERT 18 0.9 0 1.2
DIRECTION 0 0 0
INSERT 19 0.9 0 1.35
DIRECTION 0 0 0
INSERT 20 0.9 0 1.5
DIRECTION 0 0 0
EXIT MONITOR_POINTS

SINGLE_FIELD_SCALAR_TIME_OUTPUT

. .
NT 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
NQRAD 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
.
EXIT SINGLE_FIELD_SCALAR_TIME_OUTPUT

Numerical parameters:

- Simulation duration: 20 s
- $\text{CFLC} = 20$, $\text{CFLV} = 2$ (defaults)
- Compressible solver (advanced settings, default)

Boundary conditions:

- NOZZLE type boundary conditions for all boundaries (default)
Initial conditions:
  - Standard initial conditions in FLACS-CFD

Leaks:
  - Type of leak: point jet leak
  - Number of leak points: 1
  - Direction: -X
  - Total Leak Area: 0.000685943 m² (effective diameter=29.55 mm)
  - Mass Flow rate: 0.3 kg/s
  - Relative turbulence intensity: 0.1 (10 %)
  - Turbulent length scale: 0.002955 m (∼10% D)
  - Temperature: 36.14°C

Radiation:
  - Regarding radiation modelling with the Discrete Transfer Method (DTM)
    - keep the defaults for Radiation model (Discrete Transfer Method),
    - set Radiation start to When Ignited
    - keep the default for Radiation start ramp (0)
    - set Emissivity: 0.8
    - keep the defaults for Absorption coefficient model ("DANISH-COUPLED-WSGGM"), Constant scattering coefficient ("0"), DTM loop max ("25"), DTM tolerance ("0.001"), DTM frequency: iteration period ("100"), DTM frequency: time period ("-1"), DTM ray distribution ("Staggered"),
    - set DTM number of rays: 100,
    - keep the defaults for DTM verbose (0), DTM full flux (0), DTM buffer size ("-1 0.05 -1")
    - keep the far-field model defaults for far field radiation (1), far field shadow (1)
    - keep the DTM domain constraints: Automatic

Combustion:
  - Combustion model: "EDDY DISSIPATION CONCEPT (EDC)"

Soot:
  - Soot Model: Formation - Oxidation

Leaks:
  - Conduction type: NO CONDUCTION

Results:
The figures below show flame temperature and heat flux at box like target for the SINTEF impinging jet fire case.

Figure 7.26: 3D plots of temperature in SINTEF impinging jet fire case with default settings

Figure 7.27: 3D plots of total heat flux

Figure 7.28: 3D plots of radiative heat flux
7.8 Fire simulation

7.8.5.2 U.S. CG China Lake Tests, LNG Pool Fire

Problem description:

This large-scale experiment consists of an LNG pool on the sea surface (Raj, 1979; Schneider, 1979; Schneider, 1980). Measurements and observations were done for radiative heat flux, burn rate, flame geometry, and flame speed. Other large-scale experiments have also been performed, both on water and land (Luketa-Hanlin, 2006). A principal difference between LNG pool fires on land and water is the height of the flame, which is about a factor of two greater on water than on land. The reported flame length in the experiment was between 25-55 meters.

- Leakage of LNG, with composition 100% methane, on the sea modelled as a jet area leak.
- The leakage rate starts at 0.353 kg/s and increases to 35.3 kg/s during the first second, then it stays at 35.3 kg/s for the next 99 seconds (total simulation time 100 seconds). The centre of the leak is located at position x = 0 m, y = 0 m and z = 0 m. The leak has a diameter of 15 m. The reported evaporation rate of fuel from the sea surface is between 0.18 and 0.495 kg/(m² s). In this example a value of 0.2 kg/(m² s) was chosen.
- There is no wind field.
- The results of interest are the size of the flame and radiation levels.

Grid:

First a simulation volume must be defined. There is no wind field in this simulation, hence the domain in the XY-plane should be equally spaced around the leak. In the X- and Y-direction, the domain should cover the area of the pool surface plus twice the pool diameter to each side of the area covering the pool. In the Z-direction, the simulation volume should include 6-7 times the diameter of the pool. The following simulation volume is used in this example: xmin = -45, xmax = 45, ymin = -45, ymax = 45, zmin = 0, zmax = 100. Finer grids give more accurate results (cf. Coupled pool fires for grid guidelines) and it is recommended to use a grid spacing smaller than 1 meter in the X- and Y-direction in the area of the leak. Use square cells. It is not easy to know in advance how high and wide the flame will develop, so in order to avoid problems on the boundaries, the simulation volume should be made sufficiently big (see above for guidance). A pool fire will have so-called puffing. To resolve this, the numerical grid needs to be sufficiently fine (with a too coarse mesh the flame will have a more cylindrical shape).
In this example, a grid can be chosen as follows: In the X- and Y-direction, initialise a 0.5 meter grid. From the boundary of the leak area surface (Z=0), stretch the grid to the boundaries. For the Z-direction, start with a grid spacing of 0.5 meters and stretch the grid from Z=50 meters. Keep a maximum grid spacing of 2 meters to the upper boundary. The grid has now about 590000 control volumes and the porosities must be calculated. Remember to save the scenario.

Simulation and output control

- Simulation duration: 100 s
- CFLC = 20, CFLV = 2 (defaults)
- Compressible solver (advanced settings, default)

Boundary conditions:

- NOZZLE type boundary conditions for all boundaries (default)

Initial conditions:

- Standard initial conditions in FLACS-CFD

Gas composition and volume

- Methane 100% (volume fraction 1)

Ignition

Set an ignition volume with dimension 10×10×20 meters in X-Y-Z direction at the centre of the XY-plane and from Z=0.

Leaks and leak file

Set up an area leak with a diameter of 15 meters, with the leak pointing upwards. Gaussian profile and elliptical shape, 0.2kg/(m² s).

Radiation:

Regarding radiation modelling with the Discrete Transfer Method (DTM)

- keep the defaults for Radiation model ("Discrete Transfer Method"), Radiation start ("Immediately"), Radiation start ramp ("0"),
- set Emissivity: 0.8
- keep the defaults for Absorption coefficient model ("DANISH-COUPLED-WSGGM"), Constant scattering coefficient ("0"), DTM loop max ("25"), DTM tolerance ("0.001"), DTM frequency: iteration period ("100"), DTM frequency: time period ("-1"), DTM ray distribution ("Staggered"), DTM number of rays ("100"), DTM verbose ("0"), DTM full flux ("0"), DTM buffer size ("-1 0.05 -1")

Combustion:

- Combustion model: "EDDY DISSIPATION CONCEPT (EDC)" (default)
- FIRESWITCH=2

Soot:

- Soot Model: "NONE"

Conduction:

- Conduction type: "NO CONDUCTION" (default)

Results:

The figure below shows a temperature plot from the simulation of the U. S. CG China Lake tests. The results shown here account for radiative heat loss. The published flame height is between 25 and 55 meters, and the model results are in the same range.
7.8 Fire simulation

Figure 7.30: Temperature plot of the U.S. CG China Lake tests LNG pool fire.

Discussion:

- Good agreement between the measurements and numerical results is obtained.
- The results are sensitive with respect to grid refinement. A too coarse grid will not catch the physics of the puffing.

7.8.6 Source term modelling for pool fires

The default model in FLACS-CFD for pool fires is the “uncoupled pool fire model”. In the uncoupled pool fire model it is assumed that there is no feedback from the pool fire on the burning rate. A uniform burning rate is instead prescribed based on empirical correlations. The uncoupled pool fire model is most suitable for pool fires where the assumption of a uniform fixed evaporation rate holds, i.e. in relatively open areas, and for pools that remain approximately circular. For other cases and especially for under-ventilated scenarios, it is recommended to use the coupled pool fire model.

For this model, FLACS-CFD prescribes the burning rate based on the mixture, the effective pool diameter, and wind conditions. The table below lists the burning rates used based on (Rew, P.J., Hulbert, W.G. (1996)). If the burning rate is not tabulated, FLACS-CFD employs the Burgess correlation from the Yellow Book for predicting the mass burning rates of uid flammable materials under ambient conditions (see EFFECTS user and reference manual). The ultimate values are then corrected for both the pool size and wind speed (Bubbico et al. (2016)).

The table below lists typical evaporation/burning rates for a selection of flammable liquids for large pool fires, based on (Rew, P.J., Hulbert, W.G. (1996)).

Table 7.3: Summary of typical evaporation/burning rates for selected liquids (Rew,1996)

<table>
<thead>
<tr>
<th>Material</th>
<th>Mass burning rate [kg/(m^2 s)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acetone</td>
<td>0.038</td>
</tr>
<tr>
<td>Benzene</td>
<td>0.085</td>
</tr>
<tr>
<td>Butane</td>
<td>0.11</td>
</tr>
<tr>
<td>Crude oil</td>
<td>0.051</td>
</tr>
<tr>
<td>Diesel</td>
<td>0.054</td>
</tr>
<tr>
<td>Ethane</td>
<td>0.141</td>
</tr>
<tr>
<td>Ethanol</td>
<td>0.02</td>
</tr>
<tr>
<td>Material</td>
<td>Mass burning rate [kg/(m²·s)]</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>Fuel oil</td>
<td>0.034</td>
</tr>
<tr>
<td>Gasoline</td>
<td>0.067</td>
</tr>
<tr>
<td>Heptane</td>
<td>0.081</td>
</tr>
<tr>
<td>Hexane</td>
<td>0.075</td>
</tr>
<tr>
<td>Hydrogen (liquefied)</td>
<td>0.161</td>
</tr>
<tr>
<td>JP4</td>
<td>0.056</td>
</tr>
<tr>
<td>JP5/kerosine</td>
<td>0.063</td>
</tr>
<tr>
<td>LEG</td>
<td>0.14</td>
</tr>
<tr>
<td>LNG/Methane</td>
<td>0.141</td>
</tr>
<tr>
<td>LNG/Methane (water)</td>
<td>0.282</td>
</tr>
<tr>
<td>LPG/Propane</td>
<td>0.118</td>
</tr>
<tr>
<td>LPG/Propane (water)</td>
<td>0.256</td>
</tr>
<tr>
<td>Methanol</td>
<td>0.02</td>
</tr>
<tr>
<td>Naptha/Pentane</td>
<td>0.095</td>
</tr>
<tr>
<td>Octane</td>
<td>0.081</td>
</tr>
<tr>
<td>Toluen</td>
<td>0.066</td>
</tr>
<tr>
<td>Xylene</td>
<td>0.09</td>
</tr>
</tbody>
</table>

### 7.8.7 Coupled pool fires

The main difference when running the coupled pool fire model in FLACS-Fire is the way evaporation is modelled. In the uncoupled pool fire model the evaporation rate is either read off from a table or approximated via an empirical correlations. For the coupled pool fire model the pool model calculates the evaporation from the pool by a heat transfer balance including heat from the flame. The pool setup parameters must be specified correctly. Ignition must be set later than the startup time for the pool if persistent ignition is not active, that is $\text{DURATION_OF_IGNITION} = 0$.

The results are grid sensitive. A fine grid in the vicinity of the pool surface is required to anchor the flame properly. It is recommended to do some grid sensitivity studies for the area close to the pool surface to check that the physics of the fire is captured. In the example pool spread simulation a sensible grid is described. When running with the coupled pool fire model it is recommended to use cubical cells close to the pool surface. The mixing region above the pool, in the z-direction, is sensitive to the grid resolution. About 1 m above the surface the grid may be stretched to bigger cells, but keep cell sizes within 1 m. Above the assumed flame height the grid may be stretched further.

The computational domain should be at least two pool diameters to each side of the leak in the XY-plane. If a wind field is applied, then the domain should be at least 4 pool diameters in the downwind direction. In the vertical direction, the domain should be at least 6 pool diameters. Since pool fires result in puffing (oscillatory behaviour of the flame), a fine grid is needed to capture this effect. It is not easy to predict what the grid spacing should be. It depends on what one is trying to accomplish. A mesh sensitivity study should always be performed; start with a relatively coarse mesh, and then gradually refine the mesh until the results show small differences. A general recommendation is to use equal-sized grid cells in the horizontal plane covering the pool area and at least 10-15 cells across the characteristic diameter of the flame $D^*$ (Lin, 2010):

$$D^* = \left(\frac{\bar{Q}}{\rho_\infty g T_\infty \sqrt{g}}\right)^{2/5}$$  \hspace{1cm} (7.1)

Here, $\bar{Q}$ [kJ/s] = pool fire heat release, $\rho_\infty$ [kg/m³] = ambient air density, $c_p$ [kJ/kg·K] = constant pressure heat capacity of the fuel, $T_\infty$ [K] = ambient air temperature, and $g$ [m/s²] = gravitational acceleration.

The spreading pool model allows the spill to move in the XY-plane. The non-spreading and spreading pool models may be used for the same study given that the spreading pool is restricted from moving by including extra obstructions in the geometry (bund).
The coupled pool-fire model has been validated for fixed size pool fires. Moving combusting spills is possible to model through pool model 3, but this is still under validation, hence no recommendations for the grid are given here.

### 7.8.8 Converting an existing dispersion scenario to a jet fire

As most of the (jet) dispersion guidelines for the time step and grid also apply for jet fire scenarios, it is straightforward and quick to convert an existing FLACS-CFD dispersion scenario to a FLACS-Fire scenario by the following steps:

1. Open the existing dispersion scenario.
2. Change template to Fire in CASD.
3. Accept reverting to template defaults [this resets Heat Switch, CFL numbers etc. to the template defaults].
4. Optionally halve the CFL numbers (to 10/1) to increase stability.
5. Change the Monitor point output: Select relevant monitor points and add variables:
   - QRAD
   - Q
   - T
   - SOOT
6. Add 3D output variables:
   - QRAD
   - Q
   - T
   - PROD
   - SOOT
7. Optionally change the DTPlot settings.
8. Add an ignition region (e.g., +/-20m in all directions from the leak point)
9. Set the ignition time (e.g., to 0.1s after leak starts to ensure there is sufficient gas between LFL and UFL to ignite).
10. Optionally set the ignition duration above zero (e.g., to 0.15s).
11. Optionally reduce the simulation time if too long (fire simulations are slower and often achieve quasi-steady state sooner than dispersion simulations. Typical values for the maximum time can be 20s–120s).
12. For radiation, the default settings are normally fine, but in some cases it may be useful to change the behaviour, for example,
   - (a) DTM_MOD_TIME = 1 (DTM frequency: time period)
   - (b) DTPlot = 1 (for videos showing radiation contours it is best to ensure DTM and other output (DTPlot) have same frequency
   - (c) DTM_MOD_ITER = -1 (DTM frequency: iteration period)

   to force DTM-recalculation and result data export once per simulated second. Setting the DTM_MOD_TIME and DTPlot to the same value will synchronise the DTM calculations with the 3D output in the video and look best when visualising 3D radiation values in a video.
7.8.9 Recommended settings in Flowvis for FLACS-Fire

The visualisation of flames and smoke has been greatly simplified in FLACS-CFD 21.3. with automatic defaults for both and it is recommended to use those settings.

7.8.10 FLACS-Fire: Considerations for optimal calculation speed

For FLACS-CFD ventilation, dispersion and explosion simulations calculating times are a function of, among others, the number of grid cells, the time step length and the simulated duration. Fire simulations are very similar to dispersion simulations, however they take longer for the following reasons:

- smaller time steps and iterations per time step may be required to obtain a converged solution,
- the DTM ray tracing routines are an extra overhead,

For identical simulation durations; fire simulations will take longer to run than comparable dispersion simulations; typically about 1.2x to 10x longer. However, often the time of interest for fire simulation can be much shorter than for dispersion simulation (e.g. when only flame position or quasi-steady state heat radiation results are needed), in which case the time to run simulations may still be less than for similar dispersion simulations.

This section aims at giving a better understanding of the factors that influence calculation times and how changing various DTM settings will impact the calculation times and to lesser extent memory footprint. In most cases it should be possible to obtain accurate results with FLACS-Fire within a few hours, up to a few days for very large simulations.

Note:

The RADI_DTMMHy_Total line in the FLACS CALL PROFILE section in the simulation log file shows total time and relative time used by the DTM routines. In cases where the DTM part of the calculation in the simulation takes a significant amount of time (e.g. more than 25%-50%, see note below) the following steps can be taken to reduce calculation time:

7.8.10.1 DTM Ray number

The time needed for the DTM part of the calculations, scales linearly with the number of rays defined in DTM_RAYS setting in the cs-file. Fewer rays will result in coarser heat radiation results within the DTM domain (e.g. more flower pattern effect), but shorter calculation times. In most cases the default number of rays will be optimal.

7.8.10.2 DTM iteration frequency

The time needed for the DTM part of the calculations, scales linearly with the frequency of DTM calls in the DTM_ITER or DTM_TIME setting in the cs-file. Less frequent DTM iteration rays will result in capturing less of the dynamic behaviour of the flames, but shorter calculation times. In most cases the default number of DTM iterations will be optimal.

7.8.10.3 DTM Domain

The size of the DTM domain influences the calculation time significantly. With the default settings, the DTM domain size is automatically adjusted around the flame, this means that when the flame gets larger the time needed for the DTM calculations will also increase. When the DTM domain size has been set manually, reducing the DTM domain when possible will help reduce calculation time.

7.8.10.4 Incompressible vs. Compressible

For most FLACS-Fire simulations there currently is very little benefit of using the incompressible solver. In addition, most of the validation has only been done for the compressible solver. It is therefore recommended to only use the compressible solver for fire simulations.
7.8.11 Choosing the optimal DTM domain size and far field settings

FLACS-Fire can limit the calculation of the DTM radiation model to only a part of the total simulation domain. This allows simulations to run much more quickly, and in combination with the optional far-field model generates smoother contours in the far field. For most scenarios, the default settings should give an optimal balance between calculation time and accuracy and validity of the results. However, in some cases you might wish to deviate from the default settings. Below is a summary of the relevant options and information for which situations they are most suitable. Note that these settings are only available when enabling the Advanced options in the radiation tab (right-click to enable).

Possible settings of DTM_DOMAIN_CONSTRAINTS:

- **AUTOMATIC** [default]: This is the default setting, it is suitable for most scenarios.
- **AUTOMATIC_MIN**: This setting is identical to the automatic setting, but keeps the DTM domain bigger or equal to a user-defined minimum size in one or more dimensions. It can be used in situations where detailed DTM results are required in a specific region, independent of the position and shape of the flame. An application would be to define the module dimensions as the minimum DTM domain, to get more accurate radiation results on all the module walls and equipment. Another application could be to force using the DTM model for radiation calculation on the ground directly under the flame, especially upward vertical flames. You can achieve this by defining the grid cells directly above the ground in Z direction as the minimum DTM domain. When defined like this, the automatic functionality will dynamically adjust the DTM domain in both the horizontal and upwards direction to optimize calculation times.
- **FULL**: By prescribing FULL, the DTM calculations will be carried out for the whole CFD domain. Simulations with the FULL setting will be much slower and use significantly more memory than either of the automatic settings. Furthermore, you will likely have to increase the number of DTM rays to get sufficient resolution in the far field and limit the DTM ray effect; this will further increase the calculation time and memory footprint. FULL can be used for very complex or detailed geometries or for backwards compatibility with FLACS v10.5.

For both automatic settings, the DTM domain will be limited and the DTM ray effect reduced; typically 100 DTM rays will be sufficient.

FAR_FIELD_MODEL

- **ENABLED** [default]: When one of the automatic DTM domain constraint models has been selected, a far-field model can be used to calculate results outside the DTM domain. In most cases this is recommended and generally the required additional calculation time will be low.
- **DISABLED**: If the results in the far-field are not at all of interest (e.g. for an offshore platform) and the AUTOMATIC_MIN setting is used to force DTM results in the areas of interest, the far field model can be disabled to reduce the calculation time and potentially reduce minor artefacts where the DTM and far field results overlap.

7.9 Equivalent stoichiometric gas cloud

7.9.1 General Principles

For a dispersion study following a leak in a process area, the main parameter is the size of the flammable gas cloud. To evaluate the hazard of a given gas cloud, Gexcon has developed methods for natural gas that aim at estimating an equivalent stoichiometric gas cloud with comparable explosion consequences; these methods have been developed in order to reduce the number of simulations that need to be carried out to do a risk study, and the principles should be applicable to other gasses as well. For uncongested explosions, the initial turbulence characteristics of the gas cloud are also important. Gexcon has therefore carefully determined the most appropriate values to use when describing the initial turbulence characteristics for an equivalent stoichiometric gas cloud, see Initial turbulence for the equivalent gas cloud.
The size of the equivalent stoichiometric gas cloud at the time of ignition is calculated as the amount of gas in the flammable range, weighted by the concentration dependency of the flame speed and expansion. For a scenario of high confinement, or a scenario where very high flame speeds (faster than the speed of sound in cold air) are expected (either large clouds or very congested situations), only expansion-based weighting is used (denoted as $Q_8$). For most situations lower flame speeds are expected and the conservatism can be reduced. Here a weighting of reactivity and expansion is used (denoted as $Q_9$). The definitions of the $Q_8$ and $Q_9$ equivalent volumes are given in Section $Q_8$ and $Q_9$ output.

The $Q_9$ cloud is a scaling of the non-homogeneous gas cloud to a smaller stoichiometric gas cloud that is expected to give similar explosion loads as the original cloud (provided the shape and position of the cloud are chosen conservatively, as is the ignition point). This concept is useful for QRA studies with many simulations, and has been found to work reasonably well for safety studies involving natural gas releases (NORSOK, 2001).

The $Q_9$ concept has also been applied to hydrogen systems for the FZK workshop experiments and has been found to give reasonably good predictions. Details can be found in (Middha et al., 2006; Middha et al., 2008).

### 7.9.2 Shape of the equivalent gas cloud

As a practical guideline, it is recommended to choose the shape of the cloud such that it will give the maximum travel distance from the ignition location to the end of the cloud for smaller clouds. For larger clouds, end ignition scenarios with longer flame travel should also be investigated. The cloud should be made as a hexahedral box with assumed “planes of symmetry” towards confinement. The aspect ratio for a free cloud should be 1:1:1, for a cloud towards the ceiling 2:2:1, towards the ceiling and one sidewall 2:1:1, etc. For a free jet in a less confined situation, the jet momentum will usually dominate the mixing of the jet until the fuel concentration has become lean, unless the wind is very strong. The cloud should be assumed to be located a small distance downwind of the jet, if possible conservatively towards obstructions/walls. For highly buoyant gases such as hydrogen, it can always be assumed that a confined or semi-confined cloud is located near the ceiling or below any other horizontal confinement (a possible deviation from this may be large liquid hydrogen releases in hot and dry surroundings).

### 7.9.3 Initial turbulence for the equivalent gas cloud

The initial turbulence conditions are likely to have a very small effect for strongly congested simulations since turbulence will be generated quickly by the geometry in the explosion region. However, for explosion scenarios with little congestion near the ignition location, it is important to describe the initial turbulence conditions appropriately. The settings in the table below are recommended if the turbulence characteristics of the gas cloud are not known (these are the default values for gas explosion simulations). The turbulence length scale is proportional to the diffusion of the cloud, so if it is set too high in the initial conditions then it is possible that parts of the cloud that are distant from the ignition will have dissipated before the combustion reaches them. It is therefore recommended that the initial turbulent length scale is not set to be higher than the default value, even in cases for which it is known to be higher in reality. The turbulence in a cloud that has some initial turbulence but has a characteristic velocity of 0 will reduce very quickly. It is therefore important that ignition happens immediately if it is intended that the cloud has turbulence when it is ignited (see Ignition of the equivalent gas cloud for details about setting the ignition appropriately).

Note that while 200 m/s is a reasonable estimate for the velocity of a jet release, FLACS-CFD is sensitive to the product of the characteristic velocity and the relative turbulence intensity, rather than to the characteristic velocity itself.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characteristic velocity</td>
<td>200 m/s</td>
</tr>
<tr>
<td>Relative turbulence intensity</td>
<td>0.3</td>
</tr>
<tr>
<td>Turbulence length scale</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 7.4: Recommended initial turbulence settings for gas explosions.
These recommended settings were calibrated using data from three experiment campaigns, including 20 explosion experiments. The figure below compares the maximum pressure measured in the experiments with the maximum pressure that was simulated using a combined dispersion-explosion simulation, (this configuration is anticipated to achieve the most accurate simulation, see Combined dispersion and explosion simulations), and using two equivalent clouds: one using the previously recommended initial turbulence settings, and one using the new recommendations. The simulated pressures are co-located with the experiment measurement locations. The x-axis is the ratio of the maximum simulated pressure to the maximum measured pressure. Where FLACS-CFD underpredicts the maximum pressure, the ratio has a value less than one, and values greater than one represent overprediction. The y-axis shows the cumulative probability for the ratio. In previous releases of FLACS-CFD, it was recommended that the initial relative turbulence intensity and turbulence length scale be set to zero. This results in much higher probability of underprediction for the maximum pressure than when a combined dispersion-explosion simulation is run. When the new recommended values are used, the probability of underprediction is much lower and is similar to the probability for combined dispersion-explosion simulations.

When the results of a dispersion simulation are available, but a combined explosion-dispersion simulation is not (for example, if a dispersion simulation was used to calculate a cloud, which was then repositioned to simulate multiple explosion scenarios in different locations), then the initial turbulence conditions for the gas cloud can be read from the dispersion simulation output. The turbulence values will vary across the cloud, and gas explosion simulations in FLACS-CFD currently assume uniform initial turbulence conditions. In most cases, it is appropriate to set the initial turbulence for the explosion simulation to be equal to the maximum turbulence conditions from the dispersion simulation output, since this will result in the most conservative predictions.

Figure 7.31: A comparison of the maximum pressures measured in experiments with those from combined dispersion-explosion simulations (‘Real cloud’), and from two equivalent cloud simulations, using the previously recommended settings, and the new recommendations. RTI and TL are the initial relative turbulence intensity and initial turbulence length scale, respectively.

7.9.4 Ignition of the equivalent gas cloud

For smaller clouds (the flame travels less than 1 m to the open boundary), ignition with maximum distance to the edges of the gas cloud is normally a conservative choice (for a cloud located in a corner, this means corner
ignition, not central ignition). This could be used to represent all scenarios. Alternatively, a distribution of several ignition points could be applied. For larger clouds, a homogenous distribution of ignition positions should be applied. It should be kept in mind that the gas clouds with the possibility of the longest flame travel are often the most dangerous ones.

For a quantitative explosion risk assessment the explosion simulations should be performed with various idealised clouds of variable size and typically using stoichiometric concentration. For the purpose of QRA, the distribution of ignition locations should be chosen to represent reality. If there are one or more highly likely ignition locations that dominate the ignition frequencies, these may be used. Otherwise, it should be assumed that a constant ignition source might lead to end ignition (where concentration reaches LFL), whereas intermittent ignition sources will be more arbitrarily distributed (with higher likelihood centrally in the cloud where concentrations are above LFL). For stratified clouds, end ignition will mean ignition in the lower end of the cloud.

Two ignition probabilities should generally be established (in case of hydrogen, the probability of spontaneous ignition should also be considered):

- \( P_{I_{\text{const}}} \) = ignition probability from constant ignition sources (per \( \text{m}^3 \) that is exposed to flammable gas for the first time in the last 1 second)
- \( P_{I_{\text{intermittent}}} \) = ignition probability for intermittent ignition sources (per \( \text{m}^3 \) flammable volume and second)

From the CFD calculations, the volume of the gas cloud with concentration between LFL and UFL will give the volume that may be ignited if exposed to an intermittent ignition source. This information should thus be related to the intermittent ignition frequencies defined. In the case of constant ignition source, the Q6 output from FLACS-CFD (or similar from another CFD tool) gives the cloud volume that was exposed to flammable gas concentrations for the first time last second. This information should be combined with the probabilities for ignition by constant ignition sources. Thus for each time step (or each 1 s) for every dispersion calculation, the probability of ignition from spontaneous, intermittent and constant ignition sources should be established, and this probability should be added to a gas cloud size class (based on corresponding Q8, Q9 or a combination).

If the gas cloud becomes rich (in a well-mixed state, but more gas available than needed to fill the volume with stoichiometric concentration) this may be represented either as a stoichiometric cloud or as a rich cloud with a slightly higher reactivity than observed. This can be done by using the flammable volume to establish the volume of the cloud and the flammable mass to establish its concentration.

### 7.9.5 Further guidelines

As described above, for scenarios with high confinement or scenarios where very high flame speeds can be achieved, one should use the Q8 value instead of Q9. To evaluate this, several cloud sizes can be simulated to identify a critical cloud size, \( Q_{\text{crit}} \), for which the flame speeds exceed e.g., 200 m/s. For clouds with \( Q_8 < Q_{\text{crit}} \), the weighting procedure above can be applied. For clouds with \( Q_8 > Q_{\text{crit}} \), one should apply the Q8 cloud as representative cloud size. For vented rooms and other situations with a significant confinement, a weighting between Q8 and Q9 volumes is suggested.

If the ventilation or a high-momentum leak (jet) creates significant turbulence in the region where ignition is expected, this turbulence should be defined as an initial condition for the CFD solver, see Initial turbulence for the equivalent gas cloud for a description of the initial turbulence settings. Further details of the approach and a discussion of potential issues can be found in (Hansen, 2011).

### 7.10 Exporting results for use in structural response software

Fire evolution and structural response are interconnected, and therefore the output from fire simulations can be useful as input to software that compute structural response. Such tools, which are often based on the finite elements method (FEM), usually support import from several file formats, and the FLACS-CFD Python API documentation includes an example script for exporting data to one of these format, i.e. the Abaqus "*.imp" format.
7.11 Setting up a blast simulation with FLACS-Blast

7.11.1 Geometry and grid

The geometry is set up as usual for FLACS-CFD, and the Grid guidelines should be followed to ensure that the grid and geometry meet the recommendations for a blast scenario.

7.11.2 Monitor points

Monitor points are set in the same manner as for other FLACS-CFD simulations, however FLACS-Blast treats all porosities as either one or zero, so monitor points should not be placed in cells where the porosity is \( \leq 0.5 \), see the blast section in Modelling and application limitations.

FLACS-Blast does not include pressure panels; to measure the pressure at walls, place monitor points between the cell centre and the adjacent wall (but not on the wall itself). The pressure measured at the monitor point will be the same as the pressure at the wall. Be aware that if the grid is refined it may be necessary to relocate the monitor points. Therefore, it is recommended that monitor points are positioned sufficiently close to the wall that they are located between the grid-cell centre and the wall for the finest expected grid resolution.

7.11.3 Simulation and output control

The default settings are appropriate for most cases. If numerical instability occurs – a rare occurrence – CFLC and CFLV may be reduced by a factor of 2. However, other possible reasons should be ruled out first. The maximum time should be set to a physical value. Usually 0.2 seconds is enough time to capture the history of the blast across the computational domain. Longer times may be required for very large domains.

7.11.4 Boundary conditions

The PLANE_WAVE boundary condition must be used with the blast simulator. See the boundary conditions section for more details. The computational domain should be such that the total volume is about 100 times larger than the volume of the initial explosive “balloon”. It is recommended to perform a sensitivity analysis to verify that the solution is independent of the domain size.

7.11.5 Explosive charge

The explosive charge is defined using the following parameters:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSITION</td>
<td>X, Y and Z locations of the explosive charge (centre of the “balloon”).</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>Weight of the explosive charge in kg.</td>
</tr>
</tbody>
</table>
| CONFINEMENT | Coordinate directions into which the blast will propagate from the initial high-pressure high-temperature region. Available options (= includes both + and - directions):  
+X, -X, =X 
+Y, -Y, =Y 
+Z, -Z, =Z  
For example, =X=Y=Z means the blast will propagate in all directions. +X−Y means the blast will propagate in the +X and -Y directions only (i.e. the −X, +Y and Z directions are blocked). |
| TYPE        | TNT or RDX.                                                                 |
7.11.6 Example: blast impact on two adjacent structures

Note:

The scenario files for this example can be found in the doc/examples/ex06_blast directory of your FLACS-CFD installation.

7.11.6.1 Problem description

The blast effects of an explosion of 0.8 kg of TNT on two adjacent structures and surrounding areas are simulated using FLACS-Blast. The structures are represented by blocks of the size 2.3 m × 2.3 m × 2.3 m. The minimum x, y and z coordinates of structure 1 are -3.45, 1.15 and 0 respectively, and for structure 2 are 1.15, 1.15 and 0. The gap between the structures is therefore 2.3 m. The TNT explosive charge is located approximately 1.1 m from structure 1. The geometric setup is presented in the figure below.

![Figure 7.32: Geometry for the blast simulation.](image)

7.11.6.2 Computational grid

The computational grid contains 1,046,032 cells. Cubical cells of 0.1 m × 0.1 m × 0.1 m are used in the region where accurate results are required, and are stretched towards the boundaries.

7.11.6.3 Monitor points

Monitor points are defined at the following locations:

<table>
<thead>
<tr>
<th>INSERT</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-2.29</td>
<td>3.46</td>
<td>1.14</td>
</tr>
<tr>
<td>2</td>
<td>-2.29</td>
<td>3.46</td>
<td>0.19</td>
</tr>
<tr>
<td>3</td>
<td>-1.14</td>
<td>2.29</td>
<td>0.19</td>
</tr>
<tr>
<td>4</td>
<td>-2.29</td>
<td>1.14</td>
<td>1.14</td>
</tr>
<tr>
<td>5</td>
<td>-2.29</td>
<td>1.14</td>
<td>0.19</td>
</tr>
<tr>
<td>6</td>
<td>-3.46</td>
<td>2.29</td>
<td>1.14</td>
</tr>
<tr>
<td>7</td>
<td>-3.46</td>
<td>2.29</td>
<td>0.19</td>
</tr>
<tr>
<td>8</td>
<td>-2.29</td>
<td>2.29</td>
<td>2.31</td>
</tr>
<tr>
<td>9</td>
<td>-3.24</td>
<td>1.14</td>
<td>0.19</td>
</tr>
<tr>
<td>10</td>
<td>-3.46</td>
<td>1.36</td>
<td>0.19</td>
</tr>
<tr>
<td>11</td>
<td>2.29</td>
<td>3.46</td>
<td>1.14</td>
</tr>
<tr>
<td>12</td>
<td>2.29</td>
<td>3.46</td>
<td>0.19</td>
</tr>
<tr>
<td>13</td>
<td>3.46</td>
<td>2.29</td>
<td>0.19</td>
</tr>
<tr>
<td>14</td>
<td>2.29</td>
<td>1.14</td>
<td>1.14</td>
</tr>
<tr>
<td>15</td>
<td>2.29</td>
<td>1.14</td>
<td>0.19</td>
</tr>
<tr>
<td>16</td>
<td>1.14</td>
<td>2.29</td>
<td>1.14</td>
</tr>
<tr>
<td>17</td>
<td>1.14</td>
<td>2.29</td>
<td>0.19</td>
</tr>
<tr>
<td>18</td>
<td>2.29</td>
<td>2.29</td>
<td>2.31</td>
</tr>
<tr>
<td>19</td>
<td>-0.01</td>
<td>-4.01</td>
<td>0.01</td>
</tr>
<tr>
<td>20</td>
<td>-5.74</td>
<td>2.29</td>
<td>0.01</td>
</tr>
<tr>
<td>21</td>
<td>-6.89</td>
<td>2.29</td>
<td>0.01</td>
</tr>
</tbody>
</table>

---

FLACS-CFD Best practice

FLACS-CFD v22.2 User’s Manual

GEXCON
7.11 Setting up a blast simulation with FLACS-Blast

At all monitor points, the following variables are recorded: NPMAX, NMIN, NPTIMP, NP, NIMP.

7.11.6.4 Single field 3D output

The following variables are recorded for the 3D output: NPMPM, NPMAX, NMIN, NPTIMP, NP, NIMP, NVVEC, NU, NV, NW.

7.11.6.5 Simulation and output control settings

TMAX 0.1
CFLC 0.025
CFLV 0.025
DTPLOT 0.005

7.11.6.6 Boundary conditions

PLANE_WAVE (default) is used on all boundaries.

7.11.6.7 Initial conditions

The default settings are used.

7.11.6.8 Explosive

POSITION -4.61 2.31 0.19
WEIGHT 0.8
CONFINEMENT =X=Y=Z
TYPE TNT

7.11.6.9 Results

The pressure history for monitor points 6, 7 and 10 (front of the structure closest to the blast) are presented below. This region experiences the maximum impact of the explosion with overpressures of over 25 bar.
Figure 7.33: Pressure history for monitor points 6, 7 and 10.

The plots below show the pressure history on the top and back of the same structure. Here the maximum pressures are under 1 bar.

Figure 7.34: Pressure history for the top (left) and back (right) of structure 1.

Next, the pressure history on the sides of the same structure and for the monitor points located opposite to the structures are displayed. On the sides, the maximum pressure is under 1 bar, while monitor point 20 records a maximum overpressure of over 12 bar.
7.11 Setting up a blast simulation with FLACS-Blast

Figure 7.35: Pressure history for the sides of structure 1 (left) and opposite the structures (right).

The results for monitor points on the structure behind the first box are shown below. The overpressures for this structure are considerably lower than for the first box, with a maximum level of just over 0.25 bar. This is consistent with the fact that this structure does not experience the full impact of the blast. Also the overpressures for locations on the side of the structures are shown below.

Finally, the pressure contours between 0 and 1 bar at 5 ms are presented, showing the shape and size of the blast wave.
7.12 Modelling and application limitations

As noted in the section on the application areas of FLACS-CFD, it is important to be aware of assumptions and limitations that are inherent to the models used for simulating flow phenomena. This section identifies such characteristics for the FLACS-CFD software.

7.12.1 Geometry modelling

It is important to include sufficiently fine details in the geometry model, for a typical industrial installation or oil/gas rig on a scale of \( \sim 100 \text{m} \) typically down to about 1-inch piping and even smaller dimensions for pipe racks, cable trays, etc. For modern process plants, it is often possible to import the geometry model from existing computer aided design (CAD) models. However, for older facilities, or during the design phase of new ones, safety engineers must seek other solutions.

There are several known issues and inherent limitations associated with the representation of complex geometries in FLACS-CFD, including:

- It may not be straightforward to import a CAD model, since there can be issues with unit conversion, inclusion of non-physical objects (e.g. tags), etc. It is therefore important to verify the geometry model before running simulations.

- Import from CAD will normally not work well for internal geometries, i.e. inside process equipment, such as silos, mills, filters, pipes, ducts, etc.

- Rotated walls (i.e. not aligned with the coordinate axes) should be avoided, see the Note about non-orthogonal elements and general objects.

- **Left difference** always subtracts 100%, even if non-zero porosity is defined.

- It is important to represent the vent openings of a semi-confined geometry properly. If FGC (Flacs Geometry Calculator) adjusts the position of objects close to the outer boundaries to match the grid, the effective vent area may be affected. You should check the representation of the vent openings by verifying the porosity using Flowvis.
• A unified mesh/primitive geometry model is stored in FLACS-CFD. It is important to verify that the geometry model and porosities are correct before starting simulations.

• FLACS has been validated for both primitive and mesh objects. However, each are treated slightly different in FGC, so results with each approach will be similar but may not give exactly the same results.

• FGC does not move large mesh objects to grid lines, which may cause additional differences. It is important to verify porosities especially for large mesh objects that should be closed.

7.12.2 Release, dispersion and wind

There are several known issues and limitations associated with the simulation of release and dispersion scenarios in FLACS-CFD and FLACS-Dispersion:

• With the achievable grid resolution in a typical FLACS-CFD simulation, it is not possible to resolve all details of complex release scenarios, such as high-momentum compressible jets or flashing liquid releases. For this reason, FLACS-CFD includes utility programs that allow such releases to be represented as pseudo sources:
  – Jet utility program,
  – Flash utility program.

• Scenarios with an unstable atmosphere (Pasquill classes A-C) are more sensitive to the choice of time step (CFL number) than scenarios with a stable atmosphere (Pasquill classes D-F). It is recommended to use the (usually) slightly more conservative Pasquill class D rather than classes A-C. Note, in some cases Pasquill class D is not more conservative than class A. For example, a horizontal passive release at a certain elevation could reach ground level if released in unstable atmosphere (A), while not in neutral atmosphere (D), since turbulence in the plume is higher for an unstable atmosphere.

• When using the wind boundary condition, it is recommended to place the area of interest at a significant distance from the inflow/outflow boundaries.

• In Flacs2, the aerodynamic roughness variable (GROUND ROUGHNESS) defines the wind profile at wind boundaries, while the ground is modelled as a smooth surface. In cases of high aerodynamic roughness or long upwind distances, it is recommended to compare wind velocity profiles just before the obstructed region with the profile at the upwind wind boundary condition. If these differ significantly, then artificial roughness blocks, defined as an array of cuboids at terrain level, can be used to mimic real roughness. In flacscfd, the aerodynamic roughness variable (GROUND ROUGHNESS) is used at solid wall boundaries corresponding to the surface defined by the terrain. The aerodynamic roughness determines the velocity and turbulence characteristics at the near-wall control volumes (see Rough wall boundary).

• Flacs uses the standard RANS k-epsilon CFD model, which assumes isotropic turbulence (e.g. turbulence is the same in X, Y, and Z direction) and therefore cannot differentiate turbulence in vertical direction separately from turbulence in the other directions. This unfortunately means it is not ideal for calculating vertical turbulence for helideck studies (e.g. the vertical turbulence criterion in CAP 437). In most cases using Flacs will be too conservative (although underprecision is theoretically also possible) and result in a much lower calculated availability for helicopter operations.

• Numerical stability (mass residual) can be an issue for certain dispersion simulations, and you may resort to various measures for removing the problem.

• The jet utility program does not provide any warning when scenario conditions differ from the assumptions that are implicit to the model. The solution calculated from the utility program represents one of several possible leak scenarios from a high-pressure reservoir: reversible adiabatic (i.e., isentropic) flow of an ideal gas from a reservoir via choked flow in the aperture; irreversible adiabatic
(i.e., non-isentropic) expansion (over-expanded jet) without air entrainment; or expansion over a normal shock to ambient pressure. These assumptions are not appropriate for very high reservoir pressures and/or temperatures, or for particle- or droplet-laden flows.

- The jet utility program should not be used for pipeline releases, for combinations of vessels and pipelines, or for liquid releases.
- The flash utility program calculates a source term after all droplets have evaporated, assuming no impingement.
- The flash program, as well as the pool model, does not account for preferential evaporation from multi-component mixtures.
- FLACS-CFD does not represent oblique jets particularly well.
- If the flow rate of a leak is specified as velocity (not mass flow) for the FAN and SUCTION features, the mass flow rate is calculated from the molecular weight of fuel rather than air.
- Since flammability limits and other combustion properties are not implemented individually for toxic species, simulations with toxic species are limited to toxicity analysis.
- Since thermodynamic properties are not implemented individually for toxic species, simulations with toxic species are limited to scenarios with no temperature differences or scenarios characterised by temperature differences but low toxic concentration (i.e. typically less than 5 vol % - 10 vol % toxic at the source).
- Laminar flows: laminar viscosity is fixed and defined by the parameter AMUL, irrespective of the species selected in the scenario. Simulation of laminar flows require setting a proper value a AMUL using KEYS in the scenario file (see Special control keys). Modelling laminar flow is not very common, an example would be low momentum releases at zero wind speed.
- Liquid pools are modelled as a two-dimensional surface which means it is not possible for the pool to occupy the same horizontal position at two different vertical levels (circumstance that can happen when the pool overflow from one level to another lower level), for more information see Setting the pool position. To overcome this restriction, in some cases it is possible to replace the evaporative release from a pool with an area leak covering the expected pool surface.
- Complex terrain is represented in the Cartesian grid adopted by FLACS-CFD as a series of solid blocks, as explained in the Terrain import section. Simulation of stratified flows, including dense gases and liquid pools, may therefore be inaccurate in complex terrain. Checking results with grid sensitivity is recommended.
- For time varying leaks (for example resulting from depressurization of reservoirs), there are additional requirements for leak type and grid refinement. For more information see Time varying leaks and the grid recommendations sections.
- FLACS-CFD relies on empirical values for the lower (LFL), upper flammability limits (UFL) and laminar burning velocities and these may not be accurate in all cases (e.g. mixtures and different initial conditions). In addition, the LFL value for Propylene and the UFL values for Acetylene, Ethylene and H2S in FLACS-CFD deviate significantly from the typical values found in literature (Lide, 2007) and are non conservative. For more details and potential impact on results see entry on LFL and UFL calculation in Gas explosions limitations.
- The default transparency settings in the Flowvis 3D plot may not be suitable for all types of visualisation. For example when visualising exact flammability or toxic limits, it is recommended to disable result transparency, use the 3D isosurface option or use the 3D (projected) surface option or visualise in a 2D plot instead.
- The Pool evaporation model was primarily developed and validated for liquified natural gas (LNG) modelling. It may therefore be less suitable for modelling other liquid types, especially when the liquid properties are very different to LNG (e.g. LH2 or gasoline).
7.12 Modelling and application limitations

7.12.3 Gas explosions

There are several inherent limitations and important guidelines associated with the simulation of gas explosion scenarios in FLACS-CFD and FLACS-GasEx, including:

• It is recommended to follow the current grid recommendations for all explosion simulations. In particular, it is important to note that applying a grid spacing of 1 cm or less in an explosion simulation is **not** recommended. Due to the nature of the sub-grid models in FLACS-CFD, convergence of results with increasing spatial or temporal resolution cannot be expected for gas explosion simulations.

• FLACS-CFD tends to underpredict the explosion pressure in unconfined congested geometries, particularly for explosion experiments featuring very high flame speeds, approaching the detonation regime. FLACS-CFD does not model the actual detonation phenomenon, but will give an indication of whether a DDT (deflagration-to-detonation transition) is likely.

• FLACS-CFD may over-predict the explosion pressure in the acceptor region for typical ‘safety gap’ studies (Gexcon, 2013).

• FLACS-CFD relies on empirical values for the lower (LFL), upper flammability limits (UFL) and laminar burning velocities. The Le Chatelier’s mixing rule is used to calculate flammability limits for mixtures. In addition, FLACS-CFD includes an alternative mixing model that uses a separate lookup table for hydrogen-nitrogen mixtures which is used for pure hydrogen/nitrogen mixtures. Errors in the flammability limits or laminar burning velocities, could result in inaccurate results, especially in calculated flammable and equivalent stoichiometric cloud volumes and explosion overpressures. When using mixtures (including inert) or custom species, it is important to verify key composition parameters reported in the log file, such as Molecular Weight (MW), LFL and UFL, are correct. For more information see the sections on hydrogen-nitrogen mixtures and Inert gas definition. In the following situations the calculated flammability limits may not be correct:
  - For mixed fuels especially at elevated pressures and/or temperatures.
  - For ”initial pressures significantly below 1 atmosphere”.
  - Mixtures with hydrogen, nitrogen and at least one more additional flammable or inert gas
  - For flammable mixtures with an oxygen content in the atmosphere that is significantly higher or lower than the standard atmospheric composition (20.95 vol% oxygen, 79.05 vol% nitrogen).
  - For fuels with high concentrations of Propylene, Acetylene, Ethylene or H$_2$S.

For further guidance and possible workarounds contact FLACS-CFD support.

• The quasi-laminar burning velocity predicted by FLACS-CFD may be too high when simulating explosions in narrow channels and pipes.

• When simulating explosions in small enclosures (volume less than 1–2 m$^3$) and narrow channels and pipes, and especially for mixtures with relatively low reactivity, it is important to activate the FLACS-CFD submodels that account for radiative heat losses.

• FLACS-CFD does not simulate the effect of acoustic instabilities, which may dominate the explosion pressure for certain scenarios involving vented empty enclosures. However, obstructions, irregular surfaces and absorbing material (often present in real applications) will significantly decrease the importance of acoustic instabilities. Further details can be found in, e.g., (Bauwens et al., 2010; Pedersen & Middha, 2012; Bauwens & Dorofeev, 2013).

• Some scenarios are particularly sensitive to the initially defined turbulence level. If a modelled gas cloud is likely to be turbulent prior to ignition, this should be set in the initial conditions, as it may be important for the severity of the subsequent explosion. See the sections on initial turbulence parameters: characteristic velocity, relative turbulence intensity, and turbulence length scale. More details about setting the initial turbulence conditions are provided in Initial turbulence for the equivalent gas cloud

---

FLACS-CFD v22.2 User’s Manual
• Although quenching due to high turbulence levels provides an upper limit to the burning velocity in FLACS-CFD, the current burning velocity model does not incorporate a parabolic dependency of the turbulent burning velocity on the rms turbulent velocity fluctuation. Hence, for applications where a high turbulence level is expected to quench the flame, the burning velocity in FLACS-CFD may be overpredicted (Arntzen, 1998).

7.12.4 Hydrogen explosions

FLACS-Hydrogen is a CFD-code for simulating dispersion and explosion scenarios with hydrogen gas in complex geometries. All functionality of FLACS-Hydrogen can be found in the full FLACS-CFD package. Important changes regarding the hydrogen combustion properties:

• Modified laminar burning velocity curve, LFL has been reduced to 4%.
• Adjustment to include the effect of the Lewis number (more than normal wrinkling on lean concentrations, less than normal wrinkling on rich concentrations).
• Stronger enhancement due to flame wrinkling (3.5 times increase in burning velocity with distance from the ignition point at stoichiometric concentration).

These changes result from extensive validation including the following experiments:

• 20m diameter hemispherical hydrogen cloud (large-scale laminar flames)
• 1.4m small-scale channel (different baffles, gas concentration, ignition location)
• 3D corner small-scale pipe arrays (different congestion, gas concentration, ignition)
• 30m tunnel FLAME facility (different venting, baffles, gas concentration)

Based on these simulation results, the performance of FLACS-CFD simulating hydrogen is considered comparable to what is generally seen when simulating other gases.

The dynamic viscosity for hydrogen in FLACS-CFD with default settings may be somewhat too high. This will only be important in situations with absolutely no turbulence, e.g., in closed vessels with no ventilation and weak temperature gradients. Upon discovery of this issue it was decided not to change the default values; instead it has been made possible to define the constant for dynamic viscosity manually in a setup-file or using the KEYS field in the scenario file. The default constant for dynamic laminar viscosity in FLACS-CFD is 2.0e-5, a more appropriate value is 0.6e-5. If needed, it may be changed using a setup file, as shown below:

```
VERSION 1.1
$SETUP
  KEYS="AMUL=Y:0.6e-5"
$END
```

A sensitivity study has shown that simulation results usually are relatively insensitive to changing AMUL. Regarding time stepping when simulating hydrogen, the standard guidelines should be applied, i.e. CLFC=5 and CFLV=0.5

7.12.5 Detonation modelling

FLACS includes a model for predicting overpressures after transition to detonation. For hydrogen this transition is automatically triggered based on the DPDX criterion. For other gases, where detonation is less likely, it must be enabled manually by the user. The following should be noted:

• The automatic detonation transition criterion has been selected based on available hydrogen validation cases to best balance false negatives and false positives, with slight conservative bias to prevent false negatives;
• Less validation data is available for DDT transition of other gases and those results are therefore more uncertain;

• FLACS does not fully resolve the detonation shockwave, but uses an engineering model to correct for this.

7.12.6 Jet and pool fires

There are several inherent limitations and important guidelines associated with the simulation of jet and pool fires in FLACS-Fire, including:

• FLACS-Fire may under predict flame lift-off length for jet fires. For more information see section on underprediction of lift-off flame length.

• FLACS-Fire may overpredict flame buoyancy for horizontal hydrogen jet flames. In most cases the overprediction is modest.

• FLACS-Fire by default uses a different radiation model for the near field (DTM) zone and far field zone. In the overlapping area the results are merged and may occasionally result in some minor artefacts when visualising.

• FLACS-Fire may significantly overpredict (up to 2.5 times) far field radiative heat flux for monitor points which are closest and in line with non-impinging jet fires. As the far field radiative heat flux results are combined with the DTM radiative heat flux results by default, this may also cause overprediction in the non far-field radiative heat flux and total heat flux output variables. For these type of cases, it is recommended to expand the DTM domain to cover the volume of interest and disable the far field radiation model, which will mitigate this issue.

• The limitations in the pool-fire model are similar to the limitations in the regular pool evaporation/dispersion model, for more information see section on dispersion limitations.

• The pool-fire model has been validated for a modest range of scenarios, however the validation is much more limited than for jet-fires, dispersion and gas explosion scenarios. When running pool-fire simulations, it is therefore recommended to do additional simulations to assess grid sensitivity and carefully review results.

7.12.7 Dust explosions

There are several known issues and inherent limitations associated with the simulation of dust explosion scenarios with FLACS-DustEx, including:

• It is recommended to follow the current grid recommendations for all explosion simulations.

• The modelling approach in FLACS-CFD DustEx is limited to organic materials with a size-corrected rate of pressure rise (KSt value) exceeding 100–120 bar m/s.

• FLACS-DustEx includes a simple model for dust lifting (Skjold, 2007b; Skjold et al., 2007), but the model requires empirical input, and does not account for phenomena such as dust settling, separation processes in cyclones, etc.

• The results obtained for vented explosion scenarios tend to be quite grid dependent, and Gexcon recommends comparing model results with experimental results obtained with a similar dust (composition and reactivity) in experiments at similar scale.

• The results can be very sensitive to small changes in the initial conditions (Skjold, 2010).

• Validation of FLACS-DustEx is so far limited to using standard, atmospheric, oxygen fraction in air (not validated for reduced or increased oxygen content, different from standard air).
7.12.8 Blast wave propagation

In addition to the above limitations for FLACS-CFD, the following further limitations should be taken into account when simulating blast wave propagation with FLACS-Blast:

- FLACS-Blast does not simulate fragments.
- It is not possible to use panels with the Euler solver.
- FLACS-Blast does not account for the afterburning of expanding explosion products.
- FLACS-Blast treats porosities as binary, so any partial porosities are rounded to either zero or one. Cells with porosity = 0.5 are therefore ambiguous, and it is not recommended to position monitor points in cells with porosity \( \leq 0.5 \) since these will be treated as fully blocked in FLACS-Blast.

The models included in FLACS-Blast work well for far-field blast propagation; near-field results and internal explosions may be unreliable. Regarding far-field blast simulations with FLACS-CFD, Hansen et al. (2010) and Hansen & Johnson (2014) suggested a method to improve the preservation of the form of the pressure wave by a suitably timed reduction of the time step. Typically this will require a simulation with the normal time step first, and the total simulation time will be increased by a factor 4-5. Gexcon has compared the results using the proposed method with the multi-energy curves from TNO; while the pressure predictions in the far-field tend to be improved, the form of the curves is still quite different. When applying the method, one has to be careful not to reduce the time step before the pressure wave has left the flame front, as this can affect the combustion rate and change the source pressure.

7.12.9 Terrain representation

Some inherent inaccuracy will be introduced by the representation of imported terrain as blocks on the Cartesian grid. Primarily this will contribute to the generation of additional artificial turbulence near the ground. In addition, the stepped representation of the ground surface can affect the dispersion of vertically stratified flows resulting from the release of dense gases.

7.12.9.1 Dispersion

For gas dispersion the artificial turbulence will increase mixing and diffusion near the ground and thus reduce calculated dispersion results, especially in the far field (e.g., > 100 m–200 m from the release). This effect is expected to be strongest for dense plume/jet dispersion (where negative buoyancy forces dominate) with the gas cloud flowing close to the terrain level. In addition, in vertically stratified flows, the stepped representation of the terrain surface may enhance the separation of cloud layers with different densities leading to an artificial confinement of denser layers. Fewer issues are expected for passive plume/jet dispersion (where the release momentum and ambient turbulence dominate the dispersion) or for positively buoyant plume/jet dispersion. It is therefore recommended to use a terrain to represent the ground surface for passive dispersion scenarios, but, until further validation is done, it is recommended that terrains are used with caution when simulating the active dispersion of dense gases, and a grid sensitivity study is generally advisable. To determine if a plume of dense gas is passive or active, typically one of the variations of the Richardson number can be used. For full scale dense gas dispersion where the source momentum is unimportant (e.g., pool evaporation), the following criterion for passive plumes can be used (Britter, 1988):

\[
\frac{U_{\text{ref}}}{\left( \frac{g' Q_0}{D} \right)^{1/3}} > 6,
\]

where, \( U_{\text{ref}} \) [m/s] is the wind speed at 10 m height, \( g' Q_0 = g (\rho_{\text{gas}} - \rho_{\text{air}}) / \rho_{\text{air}} \) [m/s²] is the reduced gravity, and \( D \) [m] is the characteristic horizontal dimension of the leak or pool.

Note that this criterion is relatively strict, so most pool dispersion scenarios will be classified as active.
For dense jets, mixing is triggered by the release momentum instead of the ambient wind. In these cases, the densimetric Froude number replaces the Richardson number in the definition of passive conditions:

$$\frac{U_{\text{leak}}}{\sqrt{g' \cdot D}} \gg 1,$$

where, $U_{\text{leak}} \ [m/s]$ is the leak velocity, $g' = g \left( \rho_{\text{gas}} - \rho_{\text{air}} \right) / \rho_{\text{air}} \ [m/s^2]$ is the reduced gravity, and $D \ [m]$ is the diameter of the jet.

It is recommended to use a relatively fine vertical grid resolution $\Delta z$ (e.g. 20 cm – 50 cm) near the imported terrain and a corresponding horizontal resolution $\Delta x \leq \Delta z / s_x$ and $\Delta y \leq \Delta z / s_y$ based on the local slopes $s_x$ and $s_y$, respectively, in the $x$ and $y$ directions. For further recommendations regarding grid resolution, see grid_bestpractice. In line with the existing dense gas dispersion recommendation, it is also recommended to model dispersion over terrain with a maximum slope of $10^\circ$ only.

### 7.12.9.2 Explosion and fire

For explosion and fire simulations no significant issues are expected for the terrain functionality. However, for deflagrations where the gas cloud is on top of a terrain, a slight increase in explosion overpressures is expected due to the additional artificial turbulence over the terrain, but this effect is expected to be very modest. For fire simulations, the box approximation of the terrain may influence the radiation pattern and may result in local over- or under-estimation.
Chapter 8

Technical Reference

This chapter contains an overview of the theoretical foundation for the FLACS-CFD software, including physical and chemical models.

8.1 Definitions and gas thermodynamics

This section presents definitions of gas and mixture parameters and relations for ideal gases.

8.1.1 Definitions

Number of moles of a species:

\[ n_i = \frac{m_i}{M_i} \]  \hspace{1cm} (8.1)

Mole fractions:

\[ X_i \equiv \frac{n_i}{\sum_{j=1}^{N} n_j} \]  \hspace{1cm} (8.2)

Mass fractions:

\[ Y_i \equiv \frac{m_i}{\sum_{j=1}^{N} m_j} \]  \hspace{1cm} (8.3)

Fuel-oxidant ratio:

\[ \frac{F}{O} \equiv \frac{m_{\text{fuel}}}{m_{\text{ox}}} \]  \hspace{1cm} (8.4)

Equivalence ratio:

\[ \Phi \equiv \frac{(F/O)}{(F/O)_{\text{stoich}}} \]  \hspace{1cm} (8.5)

Mixtures of fuel and oxidant are characterised by the equivalence ratio as follows:

- \( \Phi > 1 \): Fuel rich mixture
- \( \Phi = 1 \): Stoichiometric mixture
- \( \Phi < 1 \): Fuel lean mixture

The mixture fraction, \( \xi \), describes the degree of mixing between two well-defined states (0 and 1) and is defined as follows:

\[ \xi = \frac{\phi - \phi^0}{\phi^1 - \phi^0}, \]  \hspace{1cm} (8.6)

where \( \phi \) is a general variable.

The progress variable \( \chi \) tells how much of the potential fuel that has burnt and is defined as follows:

\[ \chi = \frac{Y_{\text{fuel}}^0}{Y_{\text{fuel}}} + \xi \left( Y_{\text{fuel}}^1 - Y_{\text{fuel}}^0 \right). \]  \hspace{1cm} (8.7)
8.1.2 Mixing of several gases

Mole fraction:

\[ X_i = \frac{Y_i / M_i}{\sum_{j=1}^{N} Y_j / M_j} \]  
\[ (8.8) \]

Mass fraction:

\[ Y_i = \frac{X_i M_i}{\sum_{j=1}^{N} X_j M_j} \]  
\[ (8.9) \]

8.1.3 Ideal gas relations

Ideal gas law for a mixture:

\[ p = \rho RT \]  
\[ (8.10) \]

Ideal gas law for single species:

\[ p_i = \rho_i R_i T \]  
\[ (8.11) \]

Dalton’s law for a perfect gas:

\[ p = \sum_{i=1}^{N} p_i = \frac{R_u T}{V} \sum_{i=1}^{N} n_i \]  
\[ (8.12) \]

Isentropic ratio:

\[ \gamma = \frac{c_p}{c_v} \]  
\[ (8.13) \]

Speed of sound:

\[ c = \sqrt{\gamma RT} = \sqrt{\frac{p}{\rho}} \]  
\[ (8.14) \]

Mach number:

\[ Ma \equiv \frac{u}{c} \]  
\[ (8.15) \]

Pressure-density-temperature:

\[ \left(\frac{p}{p_0}\right) = \left(\frac{\rho}{\rho_0}\right)^\gamma = \left(\frac{T}{T_0}\right)^{\gamma/(\gamma - 1)} \]  
\[ (8.16) \]

8.1.3.1 Isentropic flow of an ideal gas

When assuming isentropic flow (as, e.g., in Jet), the above can also be expressed as

\[ \left(\frac{p}{p_0}\right) = \left(1 + \frac{\gamma - 1}{2} Ma^2\right)^{-\gamma/(\gamma - 1)} \]  
\[ (8.17) \]

8.2 Stoichiometric reaction

Combustion denotes the oxidation of a fuel accompanied by the production of heat and light. In most burning processes, air is the oxidant. A simple main reaction can be written as:

\[ C_{nc} H_{nh} O_{no} + aO_2 \rightarrow ncCO_2 + bH_2O + Q. \]  
\[ (8.18) \]
This reaction is stoichiometric because there is neither fuel nor oxidant left after the reaction is completed. The stoichiometric amount of oxidant on molar basis can be calculated by:

$$a = n_c + \frac{nh}{4} - \frac{no}{2}.$$  \hspace{1cm} (8.19)

The combustion products produced in the reaction are water vapour (H₂O) and carbon dioxide (CO₂). Some relations for mixing fuel with air are listed below. The mole fraction of O₂ in air is set to 20.95%, which corresponds to a mass fraction of 23.2%. This is the normal air composition in FLACS-CFD, see Initial conditions.

Stoichiometric oxidant-fuel ratio on mass basis:

$$r_{ox} = \frac{M_{O_2}}{M_{\text{fuel}}}.$$  \hspace{1cm} (8.20)

Stoichiometric air-fuel ratio on mass basis:

$$r_{air} = (1 + \frac{Y_{N_2}}{Y_{O_2}}) r_{ox} = 4.31 r_{ox}.$$  \hspace{1cm} (8.21)

Mass fraction of fuel given an equivalence ratio Φ:

$$Y_{\text{fuel}} = \frac{\Phi}{\Phi + r_{air}}.$$  \hspace{1cm} (8.22)

### 8.3 Governing equations for fluid flow

This section describes the mathematical model for compressible fluid flow used in FLACS-CFD.

Conservation of mass:

$$\frac{\partial}{\partial t} (\beta_v \rho) + \frac{\partial}{\partial x_j} (\beta_j \rho u_j) = \frac{\dot{m}}{V}$$  \hspace{1cm} (8.23)

Momentum equation:

$$\frac{\partial}{\partial t} (\beta_v \rho u_i) + \frac{\partial}{\partial x_j} (\beta_j \rho u_i u_j) = -\beta_v \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} (\beta_j \sigma_{ij}) + F_{o,i} + F_{w,i} + \beta_v (\rho - \rho_0) g_i,$$  \hspace{1cm} (8.24)

where $F_{w,i}$ is flow resistance due to walls and $F_{o,i}$ is flow resistance due to sub-grid obstructions:

$$F_{o,i} = -\rho \left| \frac{\partial \beta}{\partial x_i} \right| u_i |u_i|.$$  \hspace{1cm} (8.25)

$\sigma_{ij}$ is the Stress tensor.

Transport equation for enthalpy:

$$\frac{\partial}{\partial t} (\beta_v \rho h) + \frac{\partial}{\partial x_j} (\beta_j \rho u_j h) = \frac{\partial}{\partial x_j} \left( \beta_j \frac{\mu_{\text{eff}}}{\sigma_h} \frac{\partial h}{\partial x_j} \right) + \beta_v \frac{Dp}{Dt} + \frac{\dot{Q}}{V}$$  \hspace{1cm} (8.26)

Transport equation for fuel mass fraction:

$$\frac{\partial}{\partial t} (\beta_v \rho Y_{\text{fuel}}) + \frac{\partial}{\partial x_j} (\beta_j \rho u_j Y_{\text{fuel}}) = \frac{\partial}{\partial x_j} \left( \beta_j \frac{\mu_{\text{eff}}}{\sigma_{\text{fuel}}} \frac{\partial Y_{\text{fuel}}}{\partial x_j} \right) + R_{\text{fuel}},$$  \hspace{1cm} (8.27)

where $R_{\text{fuel}}$ is the fuel reaction rate, which will be handled in combustion modelling.

Transport equation for the mixture fraction:

$$\frac{\partial}{\partial t} (\beta_v \rho \xi) + \frac{\partial}{\partial x_j} (\beta_j \rho u_j \xi) = \frac{\partial}{\partial x_j} \left( \beta_j \frac{\mu_{\text{eff}}}{\sigma_{\xi}} \frac{\partial \xi}{\partial x_j} \right)$$  \hspace{1cm} (8.28)
Transport equation for turbulent kinetic energy:

\[
\frac{\partial}{\partial t} (\beta_v \rho k) + \frac{\partial}{\partial x_j} (\beta_j \rho u_j k) = \frac{\partial}{\partial x_j} \left( \beta_j \frac{\mu_{\text{eff}}}{\sigma_k} \frac{\partial k}{\partial x_j} \right) + \beta_v P_k - \beta_v \rho \varepsilon \tag{8.29}
\]

Transport equation for the dissipation rate of turbulent kinetic energy:

\[
\frac{\partial}{\partial t} (\beta_v \rho \varepsilon) + \frac{\partial}{\partial x_j} (\beta_j \rho u_j \varepsilon) = \frac{\partial}{\partial x_j} \left( \beta_j \frac{\mu_{\text{eff}}}{\sigma_\varepsilon} \frac{\partial \varepsilon}{\partial x_j} \right) + \beta_v P_\varepsilon - C_2 \beta_v \rho \varepsilon^2 \tag{8.30}
\]

The stress tensor in the above equations is given by:

\[
\sigma_{ij} = \mu_{\text{eff}} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \delta_{ij} \left( \rho k + \mu_{\text{eff}} \frac{\partial u_k}{\partial x_k} \right) \tag{8.31}
\]

The effective viscosity is defined as follows:

\[
\mu_{\text{eff}} = \mu + \rho C_\mu \frac{k^2}{\varepsilon}, \tag{8.32}
\]

where the second term is known as the turbulent viscosity or eddy viscosity.

Flow shear stresses, \( G_s \), wall shear stresses, \( G_w \), buoyancy, \( G_b \), and sub-grid objects, \( G_o \), contribute to the production of turbulent kinetic energy \( P_k \):

\[
P_k = G_s + G_w + G_b + G_o. \tag{8.33}
\]

The production rate of turbulent kinetic energy due to shear stresses appears from the derivation of the transport equation and reads:

\[
G_s = \sigma_{ij} \frac{\partial u_i}{\partial x_j}. \tag{8.34}
\]

Production due to buoyant forces modelled by a simple gradient model:

\[
G_b = -\frac{1}{\rho} \frac{\mu_{\text{eff}}}{\sigma_b} g_i \frac{\partial \rho}{\partial x_i}. \tag{8.35}
\]

The turbulence generation due to sub-grid obstructions is modelled by:

\[
G_o = C_o \beta_v \rho |\vec{u}| u_i^2 f_i, \tag{8.36}
\]

where \( C_o \) is a model constant and \( f_i \) is a parameter depending on sub-grid objects.

The production of dissipation, \( P_\varepsilon \), is modelled as follows:

\[
P_\varepsilon = C_1 \frac{\varepsilon}{k} P_k \left( 1 + C_3 \varepsilon R_f \right), \tag{8.37}
\]

where the model for the buoyancy term follows (Rodi, 1980):

\[
R_f = -\frac{G_b |\vec{u} \times \vec{g}|}{P_k |\vec{u}| |\vec{g}|}. \tag{8.38}
\]

In FLACS-CFD, the buoyancy terms \( G_b \) and \( R_f \) are zero when products are present.

### 8.4 Turbulence modelling

Turbulence is modelled by a two-equation model, the \( k - \varepsilon \) model. It is an eddy viscosity model that solves two additional transport equations; one for turbulent kinetic energy and one for dissipation of turbulent kinetic energy. Following Boussinesq eddy viscosity assumption, an eddy viscosity models the Reynolds stress tensor as:

\[
-\rho \overline{u_i' u_j'} = \mu_{\text{eff}} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \rho \frac{2}{3} \varepsilon \delta_{ij}. \tag{8.39}
\]
A few constants are included in the equations mentioned above. In FLACS-CFD, the following set of constants is used, which agree with the model by Launder and Spalding (1974):

\[
\begin{array}{c|cccc}
C_\mu & C_{1\varepsilon} & C_{2\varepsilon} & C_{3\varepsilon} \\
0.09 & 1.44 & 1.92 & 0.8 \\
\end{array}
\]

In addition, there is a set of turbulent Prandtl-Schmidt numbers, \(\sigma_\phi\). Prandtl-Schmidt numbers compare the diffusion of the variable in question to the dynamic viscosity. The turbulent Prandtl-Schmidt numbers are:

\[
\begin{array}{cccccc}
\sigma_h & \sigma_{\text{fuel}} & \sigma_\xi & \sigma_k & \sigma_\varepsilon & \sigma_\theta \\
0.7 & 0.7 & 0.7 & 1.0 & 1.3 & 0.9 \\
\end{array}
\]  

(8.40)

### 8.5 Wall functions

Boundary layers are regions in the flow field close to walls and obstructions where there are steep gradients and peak values for turbulent kinetic energy and its dissipation rate. Very close to the wall surface viscous forces dominate over inertial effects. The motivation for using wall-functions is to model the influence of the wall at a point a certain distance from the wall.

A dimensionless wall distance is defined by:

\[
y^+ = \frac{\rho C_\mu^{1/4} k^{1/2} y}{\mu},
\]

where \(y\) is the distance from the wall point to the wall. Wall point is defined as the point closest to the wall where transport equations are solved. The shear stresses caused by the wall are modelled by:

\[
\tau_{w,i} = \begin{cases} 
\frac{\mu u_i}{y} & \text{if } y^+ < E^+ \\
\frac{\rho u_i c_{1}\sqrt{k^2}}{E^+ + \ln\left(\frac{y^+}{E^+}\right)} & \text{if } y^+ \geq E^+.
\end{cases}
\]

(8.42)

The wall friction term in the momentum equation becomes:

\[
F_{w,i} = -\beta_v \tau_{w,i} \frac{A_w}{V}.
\]

(8.43)

Production of turbulent kinetic energy in the wall point is modelled by:

\[
G_w = \begin{cases} 
0 & \text{if } y^+ < E^+ \\
\frac{2\tau_i^2 \ln\left(\frac{y^+}{E^+}\right)}{y_{cv} \rho C_\mu} & \text{if } y^+ \geq E^+.
\end{cases}
\]

(8.44)

Dissipation of turbulent kinetic energy is given a value at the wall point by solving the following integral:

\[
\varepsilon_w = \frac{1}{y_{cv}} \int_{0}^{y_{cv}} \varepsilon dy.
\]

(8.45)

The integral is estimated by:

\[
\varepsilon_w = \begin{cases} 
\frac{1}{y_{cv}} \left( \frac{2\mu k}{y^+} + \varepsilon (y_{cv} - y) \right) & \text{if } y^+ < E^+ \\
\frac{1}{y_{cv}} \left( \frac{k y^+}{y^+ - 1} \right) - \left( \frac{k_{i+1} - k}{y_{cv} - y} \right) + \frac{C_\varepsilon^{3/4} k^{3/2}}{\kappa} \ln\left(\frac{y^+}{E^+}\right) + \varepsilon (y_{cv} - y) & \text{if } y^+ \geq E^+.
\end{cases}
\]

(8.46)

\(k_{i+1}\) denotes the value of \(k\) and \(y_{i+1}\) denotes the wall distance in the point beyond the wall point in the opposite direction of the wall. \(\varepsilon\) denotes the mean value of \(\varepsilon\) between the cell point and the control volume boundary in the opposite direction of the wall.
8.5.1 Rough wall boundary (flacscfd only)

For solid boundaries corresponding to a terrain surface, a different wall model is adopted in flacscfd, corresponding to the so called “fully rough regime” (Parente et al., 2011), in which the wall stress $\tau_{w,i}$ is defined as a function of the aerodynamic roughness $z_0$ as:

$$\tau_{w,i} = \frac{\rho u_i c_k^{1/4} k^{1/2}}{\ln \left( \frac{yc_v/2 + z_0}{z_0} \right)},$$

(8.47)

the production of turbulent kinetic energy is defined as:

$$G_w = \frac{\tau_w^2}{(yc_v/2 + z_0) \rho u_i c_k^{1/4} k^{1/2}},$$

(8.48)

and the dissipation rate is set to:

$$\varepsilon_w = \frac{C_{\mu}^{2/3} k^{3/2}}{\kappa (yc_v/2 + z_0)}.$$

(8.49)

8.6 Wind boundary

Wind boundaries reproduce the properties of the atmospheric boundary layer close to Earth’s surface. Monin & Obukhov (1954) developed a theory to explain buoyancy effects on the atmospheric boundary layer and defined a characteristic length scale:

$$L = -\frac{\rho c_p T_u u^*}{\kappa q H_s},$$

(8.50)

where $H_s$ is the sensible heat flux from the surface and $u^*$ is the friction velocity. The Monin-Obukhov length is a measure for the stability of the atmospheric boundary layer. Table Monin-Obukhov lengths and stability shows an interpretation of the Monin-Obukhov lengths with respect to the atmospheric stability (Bosch and Veterings, 1996).

<table>
<thead>
<tr>
<th>Monin-Obukhov length</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small negative, $-100 m &lt; L &lt; 0$</td>
<td>Very unstable</td>
</tr>
<tr>
<td>Large negative, $-10^5 &lt; L &lt; -100$</td>
<td>Unstable</td>
</tr>
<tr>
<td>Very large, $</td>
<td>L</td>
</tr>
<tr>
<td>Large positive, $10 &lt; L &lt; 10^5$</td>
<td>Stable</td>
</tr>
<tr>
<td>Small positive, $0 &lt; L &lt; 10$</td>
<td>Very stable</td>
</tr>
</tbody>
</table>

In FLACS-CFD, the Monin-Obukhov length is estimated by using Pasquill classes, which is a method of categorising the amount of atmospheric turbulence present. You have to specify the average wind velocity, $U_0$, a reference height, $z_{ref}$, an atmospheric roughness length, $z_0$, and the Pasquill class under Initial conditions. The velocity profile is logarithmic:

$$U(z) = \begin{cases} \frac{u^*}{U_0} \left[ \ln \left( \frac{z-z_0}{z_0} \right) - \psi_u(z) \right] & \text{if } z_0 > 0 \\ \frac{u^*}{U_0} \left[ \ln \left( \frac{z}{z_0} \right) \right] & \text{if } z_0 = 0, \end{cases}$$

(8.51)
where \( u^* \) is the friction velocity and \( z_d \) is the canopy height. \( u^* \) is generally given by:

\[
u^* = \frac{U_0 \kappa}{\ln \left( \frac{(z_{ref} - z_d) + z_0}{z_0} \right) - \psi_u(z_{ref})}
\]

(8.52)

and \( \psi_u \) is given as:

\[
\psi_u(z) = \begin{cases} 
0 & \text{for Pasquill class D} \\
2 \ln \left( \frac{1 + \xi}{2} \right) + \ln \left( \frac{1 + \xi^2}{2} \right) - 2 \arctan (\xi) + \frac{\pi}{2} & \text{for } L < 0 \\
-17 \left( 1 - \exp \left( -0.29 \frac{z}{L} \right) \right) & \text{for } L > 0,
\end{cases}
\]

(8.53)

where \( \xi = \left( 1 - 16 \frac{z}{L} \right)^{1/4} \).

Table Wind profile parameters below gives an overview of parameters that are used to calculate values for velocity, \( k \), and \( \varepsilon \) at wind boundaries. The values for \( L_s \) and \( z_s \) are taken from (Bosch and Wetering, 1996) and originate from the graphs in (Golder, 1972). Values for \( h \) are taken from (Han et al., 2000).

<table>
<thead>
<tr>
<th>Pasquill class</th>
<th>Stability</th>
<th>Boundary layer height, ( h )</th>
<th>( L_s )</th>
<th>( z_s )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Unstable</td>
<td>1500 m</td>
<td>33.162 m</td>
<td>1117 m</td>
</tr>
<tr>
<td>B</td>
<td>Unstable</td>
<td>1500 m</td>
<td>33.258 m</td>
<td>11.46 m</td>
</tr>
<tr>
<td>C</td>
<td>Slightly unstable</td>
<td>1000 m</td>
<td>51.787 m</td>
<td>1.324 m</td>
</tr>
<tr>
<td>D</td>
<td>Neutral</td>
<td>( \min \left( 500 \text{ m}, 0.3 \frac{u^*}{T_f} \right) )</td>
<td>1.0 m</td>
<td>0 m</td>
</tr>
<tr>
<td>E</td>
<td>Slightly stable</td>
<td>( 0.4 \sqrt{\frac{u^*}{f}} )</td>
<td>-48.33 m</td>
<td>1.262 m</td>
</tr>
<tr>
<td>F</td>
<td>Stable</td>
<td>( 0.4 \sqrt{\frac{u^*}{f}} )</td>
<td>-31.323 m</td>
<td>19.36 m</td>
</tr>
</tbody>
</table>

From the values in the wind profile parameters table, the Monin-Obukhov length can be calculated as follows (Golder, 1972):

\[
\frac{1}{L} = \frac{1}{L_s} \log \frac{z_0}{z_s}.
\]

(8.54)

The set of expressions for the wind boundary profiles for turbulent kinetic energy, \( k \), and its dissipation, \( \varepsilon \), proposed by (Han et al., 2000) are implemented in FLACS-CFD. Different expressions were proposed for unstable and stable/neutral boundary layers. Unstable boundary layers are caused by heat from the ground that increases the temperature of the air close to the surface. Hence, the density close to the surface is less than the density of the air above, which gives an unstable situation. The mean surface heat flux, \( \dot{q}_s \), is therefore an important parameter when the turbulence profiles at the inlet is estimated for unstable boundary layers. The inlet profiles for unstable boundary layers (A, B, and C) are:

\[
k(z) = \begin{cases} 
0.36 w^* 2 + 0.85 u^* 2 \left( 1 - 3 \frac{z}{L} \right)^{2/3} & \text{if } z \leq 0.1 h \\
0.36 + 0.9 \left( \frac{z}{h} \right)^2 \left( 1 - 0.8 \frac{z}{h} \right)^2 w^* 2 & \text{if } z > 0.1 h
\end{cases}
\]

(8.55)

and

\[
\varepsilon(z) = \begin{cases} 
\frac{w^* 3}{h} \left( 1 + 0.5 \left( \frac{z}{L} \right)^{2/3} \right)^{3/2} & \text{if } z \leq 0.1 h \\
\frac{w^* 3}{h} \left( 0.8 - 0.3 \frac{z}{h} \right) & \text{if } z > 0.1 h
\end{cases}
\]

(8.56)

where the heat velocity, \( w^* \), is given by:

\[
w^* = \left( \frac{\dot{q}_s h}{T_0 p c_p} \right)^{1/3},
\]

(8.57)
where the air properties, \( \rho \) and \( c_p \), are obtained at ambient temperature \( T_0 \) and pressure \( p_0 \). Profiles for neutral and stable boundary layers depend on the friction velocity and the Monin-Obukhov length as follows:

\[
k(z) = \begin{cases} 
6 u^* z & \text{if } z \leq 0.1 h \\
6 u^* (1 - \frac{z}{h})^{1.75} & \text{if } z > 0.1 h
\end{cases}
\]  
(8.58)

and

\[
\varepsilon(z) = \begin{cases} 
\frac{a^{+3}}{v^{+3}} (1.24 + 4.3 \frac{z}{h}) & \text{if } z \leq 0.1 h \\
\frac{a^{+3}}{v^{+3}} (1.24 + 4.3 \frac{z}{h}) \left(1 - 0.85 \frac{z}{h}\right)^{3/2} & \text{if } z > 0.1 h.
\end{cases}
\]  
(8.59)

### 8.7 Combustion modelling

The model for premixed turbulent combustion represents turbulent flame propagation as turbulent burning velocity, calculated from diffusion and reaction rate in the numerical flame zone. This entails the modelling of the diffusion coefficient and the source term in the conservation equation for the mass fraction. The model consists of several components:

1. The numerical flame model.
2. A burning velocity correlation that relates the burning velocity to the flow regime and mixture dependent variables:
   - Laminar regime
   - Quasi-laminar regime
   - Turbulent regime
3. A flame folding model that accounts for the increase burning rate due to the increase flame surface area by:
   - Sub-grid obstructions
   - Sub-grid turbulence (modelled via the turbulent burning velocity correlation)
   - Hydrodynamic instabilities, such as the Rayleigh-Taylor instabilities.

The two-way coupling between chemical reactions and the local turbulence structures is achieved using the flamelet assumption, whereby the turbulent premixed combustion is represented by an array of laminar flame structures with a finite thickness embedded in the turbulent flow field. The combustion model is based on the flamelet correlation proposed by Bray (Bray (1990)) for turbulence burning velocity and the beta flame model developed by Arntzen (Arntzen (1998)).

Ignition of a premixed cloud of fuel and oxidant may escalate to an explosion. Before escalation, a steady non-turbulent premix of fuel and oxidant will burn with a laminar burning velocity:

\[
S^0_L = S^0_L \text{(fuel, } \Phi). \]  
(8.60)

The laminar burning velocity depends on the fuel and the equivalence ratio \( \Phi \). For mixtures with fuel contents below the Lower Flammability Limit (LFL) or above the Upper Flammability Limit (UFL), the laminar burning velocity equals zero, i.e. it will not burn. In an explosion, the flame will accelerate and become turbulent. The turbulent burning velocity is much larger than the laminar one due to much better mixing of reactants and products. FLACS-CFD uses correlations for both laminar and turbulent burning velocities that origin from experimental work.

In industrial applications, the reaction zone in a premixed flame is thin compared to practical grid resolutions. It is therefore necessary to model the flame. In FLACS-CFD, the flame zone is thickened by increasing the diffusion with a factor \( \beta \) and reducing the reaction rate with a factor \( 1/\beta \). Hence, the flame model in FLACS-CFD is called the \( \beta \)-model.
8.8 Detonation modelling

8.7.1 The FLACS-CFD flame model

The flame model defines the criteria for combustion and the spatial distribution of the reaction rate across the numerical flame zone. More specifically, the flame model defines the diffusion coefficient $D$, and the reaction rate $R_F$, in $kgm^{-3}s^{-1}$, in the fuel mass fraction equation based on an input burning velocity:

$$
\frac{\partial}{\partial t} (\beta_v \rho Y_{fuel}) + \frac{\partial}{\partial x_j} (\beta_j \rho u_j Y_{fuel}) = \frac{\partial}{\partial x_j} \left( \beta_j \rho D \frac{\partial Y_{fuel}}{\partial x_j} \right) + R_F \tag{8.61}
$$

where $c = 1 - \frac{Y_F}{Y_{F0}}$; $R_F = C_{\beta R_F} S \Delta \rho \min \left[ c, 9(1-c) \right]$, \tag{8.62}

where $C_{\beta R_F}$ is a model constant, $S$ is the burning velocity (either laminar, quasi-laminar or turbulent, depending on the local flow conditions), $\Delta$ is a constant on the order of the grid-cell size, $c$ is the progress variable, $Y_F$ the fuel mass fraction and $Y_{F0}$ is the fuel mass fraction that was initially available in the specific control volume. The progress variable is tracked by solving for the fuel mass fraction. The average chemical source term involving $R_F$ is highly non-linear and is modelled starting from the eddy break-up concept (Spalding (1971)).

The diffusion coefficient $D$ for fuel comes from the transport equation for fuel and is defined as

$$
D = \frac{\mu_{eff}}{\sigma_{fuel}} \tag{8.63}
$$

Furthermore, it is possible to define a dimensionless reaction rate $W$. In the $\beta$-model, $D$ and $W$ are adjusted as follows:

$$
W^* = \frac{W}{\beta} = W \frac{I_{LT}}{\Delta g} \tag{8.64}
$$

$$
D^* = D \beta = D \frac{\Delta g}{I_{LT}} \tag{8.64}
$$

From an eigenvalue analysis of the burning velocity (Arntzen, 1998), the following relation between the diffusion coefficient $D$ and a dimensionless reaction rate $W$ is obtained for $\chi_q = 0.05$:

$$
WD = 1.37 S_u^2 = W^* D^* \tag{8.66}
$$

$\chi_q$ is the quenching limit of the progress variable $\chi$. $D^*$ and $W^*$ depend on the grid-cell size and the burning velocity as follows:

$$
W^* = c_{1 \beta} \frac{S_u}{\Delta g} \tag{8.67}
$$

$$
D^* = c_{2 \beta} S_u \Delta g \tag{8.68}
$$

The reaction rate of fuel is modelled by the following expression:

$$
R_{fuel} = -W^* \rho \min \left( \delta_H(\chi - \chi_q), \chi, 9 - 9\chi \right), \tag{8.69}
$$

where $\delta_H$ is the Heaviside step function.

8.8 Detonation modelling
The detonation model in FLACS-CFD is an engineering approach to predict directly initiated detonations and for Deflagration to Detonation Transition (DDT) scenarios. The model is based on the work by Hansen and Johnson (2015) and developed as part of a cooperation project between HYEX Safety and Gexcon. The burning velocity is set to a constant value and there is no turbulence contribution to the flame speed. The time step is reduced by a factor of 5 and supersonic flow is allowed. When the detonation is triggered, the whole flammable cloud is considered. This is a conservative approach, since the detonability range is always within the flammable range ISO/TR 15916 2015.

When the Normalized Pressure Gradient (DPDX) exceeds 1 at one control volume, the detonation model is activated automatically for hydrogen explosions. From over 200 hydrogen explosion scenarios in Gexcon's validation database, 10 % of the experiments were reported as detonations by the respective groups performing the tests. FLACS-CFD predicts DDT for 15.7% of the hydrogen scenarios, for 10.7 % of the hydrogen scenarios not classified as detonation and for 60 % of the hydrogen scenarios classified as detonation. The scenarios for which FLACS-CFD do not predict DDT and are classified as detonation in the validation database are part of experimental campaigns with scenario variations that lead to both deflagrations and detonations. Considering that detonation events may depend on small perturbations of the initial conditions (in repeated large scale experiments with the same setup detonation may happen in some of the repetitions), the performance of the prediction model based on DPDX is acceptable. Comparison of the actual and predicted detonation frequency, is relevant for probabilistic risk analysis. The percentage of predicted DDT events (15.7 %) compared to the percentage of DDT events recorded in experiments (10.7 %) indicate that the model is conservative. This result should be interpreted considering that some detonations happening in experiments may have not been recorded and that, likely, the actual percentage of DDT events is above 10.7 %.

For other fuels than hydrogen, see section DDT on how to activate detonation. DDT_TIME can be used to set the time at which DDT is detected (see section DPDX). In chapter Validation, FLACS-CFD peak pressure predictions are compared to pressure data from experimental campaigns that include detonating tests.

8.9 Modelling of jet sources

The description in this section is of general nature, for details of the jet utility program, see the section on pseudo-source models.

To model the conditions of a pressurised reservoir which is gradually emptied through a nozzle, a simple procedure that calculates the sonic flow rate through the nozzle can be used. By assuming isenthalpic expansion, it is possible to calculate the expansion of a sonic flow analytically. Further air entrainment can be accounted for using simplifying assumptions. Finally, the calculated mass flow mixes in a well-stirred reactor with a constant volume and a constant ventilation rate. Additional values for the turbulence quantities must be calculated to use the data in FLACS-CFD.

Calculation of 5 stage analytic dispersion:

1. Reservoir (stagnation)
2. Nozzle (sonic)
3. Jet (outlet)
4. Air entrainment
5. Well-stirred reactor

Let $\Theta(u, h, T, p, A)$ be a vector describing the necessary leakage parameters, then

- $\Theta_a$ refers to the ambient condition,
• $\Theta_2$ refers to the outlet condition,
• $\Theta_1$ refers to the nozzle condition and
• $\Theta_0$ refers to the stagnation condition.

Initial reservoir conditions:
- Pressure: $p_0$ is specified.
- Temperature: $T_0$ is specified
- Volume: $V_0$ is specified
- Density: $\rho_0 = \frac{p_0}{RT_0}$
- Total mass: $m_0 = p_0 V_0$
- Heat exchange coefficient: $h_{wall}$ is specified.

Reservoir conditions at time $t + dt$:
- Heat capacity at constant pressure $[J/K]: C_p$
- Total mass $[kg]: m_{0,t+dt} = m_0 - \dot{m}_1 dt$
- Temperatures $[K]: T_{0,t+dt} = T_0 - (\dot{Q}_0 + \dot{m}_1 h_1) dt / C_p$
- Wall heat flux $[J/s]: Q = h_{wall}(T_0 - T_{wall})$
- Density $[kg/m^3]: \rho_0 = m_0 / V_0$
- Pressure $[Pa]: p_0 = \rho_0 R T_0$

Nozzle (sonic) conditions:
- Effective nozzle area: $A_1$ is specified.
- Temperature: $T_1 = T_0 (2 / (\gamma + 1))$
- Pressure: $p_1 = p_0 (T_1 / T_0)^{\gamma / (\gamma - 1)}$
- Density: $\rho_1 = \frac{p_0}{RT_1}$
- Sound speed: $c_1 = \sqrt{\gamma R T_1}$
- Velocity: $u_1 = c_1$
- Enthalpy: $h_1 = c_p T_1$
- Mass flow: $\dot{m}_1 = \rho_1 u_1 A_1$

Jet (outlet) conditions:
- Velocity: $u_2 = u_1 + \frac{p_1 - p_2}{\rho_1 u_1^2}$
- Enthalpy: $h_2 = h_1 + \frac{1}{2} (u_1^2 - u_2^2)$
- Temperature: $T_2 = T_1 + \frac{1}{2} \frac{u_1^2 - u_2^2}{c_p}$
- Pressure: $p_2 = p_a$
- Density: $\rho_2 = \frac{p_2}{RT_2}$
- Effective outlet area: $A_2 = A_1 \frac{\rho_1 u_1}{\rho_2 u_2}$
- Mass flow: $\dot{m}_1 = \rho_1 u_1 A_1$

Air entrainment condition:
- Pressure: $p_3 = p_a$
- Temperature: $T_3 = T_a$
- Density: $\rho_3 = \frac{p_3}{RT_3}$
- Velocity: $u_3 = u_2 \frac{c_p}{T_3}$
- Effective area: $A_3 = A_2 \frac{\rho_1 u_1}{\rho_3 u_3}$
- Mass flow: $\dot{m}_3 = \rho_3 u_3 A_3$

Well-stirred reactor condition:
- Volume: $V_4$ is specified.
- Ventilation rate: $V_{air}$ is specified.
\[
\text{Incremented fuel mass: } m_{\text{fuel}} = m_{\text{fuel}} + \dot{m}_{\text{fuel}} df \\
\text{Incremented air mass: } m_{\text{air}} = m_{\text{air}} + \dot{m}_{\text{air}} df \\
\text{Incremented mixture mass: } m_{\text{mix}} = m_{\text{fuel}} + m_{\text{air}} \\
\text{Density: } \rho_4 = \frac{m_{\text{mix}}}{V_4 + (V_{\text{fuel}} + V_{\text{air}}) df} \\
\text{Total mass inside volume: } m_4 = V_4 \rho_4 \\
\text{Fuel mass: } m_{\text{fuel},t+dt} = m_{\text{fuel}} \frac{m_{\text{fuel}}}{m_{\text{total}}} m_4 \\
\text{Air mass: } m_{\text{air},t+dt} = m_{\text{air}} \frac{m_{\text{air}}}{m_{\text{total}}}
\]

### 8.10 Pseudo-source models in the jet utility program

A pseudo-source model, also called notional-nozzle model, is used to calculate the status of an underexpanded jet (in terms of temperature, velocity, diameter and density) at the conditions where the jet pressure is atmospheric. A common assumption in pseudo-source models is that in the transition from under-expanded conditions to the location of the pseudo-source there is no mixing of the released gas with air.

See also (Velikorodny 2012) for a comparison of different pseudo-source models with CFD and experimental data.

#### 8.10.1 Single planar shock model

The single planar shock model in the jet utility program is based on a one-dimensional model for the release of an ideal gas from a pressurised reservoir through a nozzle into an open atmosphere. The model assumes single-phase compressible (gas) flow at all stages. A single planar shock is taken into account, more complicated three-dimensional shock structures are not considered. From inside the reservoir, where the gas is initially at rest, and up to the jet exit at the nozzle at position 1 (see the figure below), the gas flow is treated as reversible adiabatic (i.e. isentropic, and hence inviscid). In general there is sonic (choked) flow at the nozzle and supersonic expansion between the nozzle and the shock, which is located at the interface between states 2 and 3 in the figure below. Between the nozzle and the shock, the expansion is modelled as an adiabatic process for a compressible gas, that is, conservation of mass, momentum, and energy apply, but no assumption about isentropic flow is made. The thermodynamic change across the shock front is not isentropic; here the Rankine–Hugoniot relations are employed, and it is assumed that the pressure just downstream of the shock front (at position 3 in the figure below) is equal to the ambient pressure. By fixing the pressure at the ambient value, the system of (in general non-linear) equations is closed, and the desired parameters can be found by an iterative solution method.

Note that the model neglects the possible entrainment of air (ambient gas) in the region up to the hypothetic nozzle at position 3, downstream of the shock front. It is the condition at position 3 (and the area of the expanded jet there) that is reported as output from the jet utility program. From the state at position 3, the general fluid dynamics solver in the core simulator Flacs can be applied, which allows to include the modelling of entrainment.

It is recommended to use the jet utility program whenever the release scenario is within the applicable range for this program (assuming, among others, the validity of the ideal gas law, which is in general expected to model accurately the behaviour of real gases at pressures up to approximately 100-150 bar at normal temperatures). If another tool is available and expected to model the jet release better than the jet utility program for the given conditions and materials, then the leak file can be manually created using
the output from the alternative tool.

![Schematic sketch of the under-expanded jet model used in the jet utility program](image)

Figure 8.1: Schematic sketch of the under-expanded jet model used in the jet utility program, defining the states where analytical models are applied; sonic conditions are assumed at the jet exit (1), the normal shock is located at the interface between states 2 and 3; from state 3 the flow is subsonic.

A large reservoir of high pressure gas at stagnant conditions with pressure and temperature is assumed to be present upstream of what is shown in the figure above. The gas flows from the reservoir into the inlet tube with mass flow rate \( \dot{m} \) and exits from the tube into the stagnant ambient environment under sonic conditions at position 1.

The conditions at the jet exit are determined assuming an isentropic expansion from the reservoir conditions to the jet exit:

\[
\frac{p_1}{p_0} = \left( \frac{2}{\gamma + 1} \right)^{\frac{\gamma}{\gamma - 1}}
\]

\( (8.70) \)

\[
\frac{T_1}{T_0} = \frac{2}{\gamma + 1}
\]

\( (8.71) \)

\[
\rho_1 = \frac{p_1 M}{RT_1}
\]

\( (8.72) \)

\[
\nu_1 = c_1 = \sqrt{\gamma RT_1}
\]

\( (8.73) \)

\[
\dot{m} = \rho_1 \nu_1 A_1.
\]

\( (8.74) \)

The quantities \( p_0, T_0, A_1 \) (orifice area multiplied by the discharge coefficient) and the gas (hence \( M \) and \( \gamma \)) are assumed to be given. Assuming adiabatic expansion between states 1 and 2, a one-dimensional
momentum balance (ignoring entrainment and viscosity) is applied to the control volume shown in the figure below.

![Figure 8.2: Schematic view of the control volume for Jet model momentum equation.](image)

The momentum balance can be written as:

$$\rho_2 u_2^2 A_2 - \rho_1 u_1^2 A_1 = p_1 A_1 - p_2 A_2 + \bar{p} \Delta A. \quad (8.75)$$

Let $\bar{p} = \alpha p_1 + (1 - \alpha) p_2$ and let $\Delta A = A_2 - A_1$; then the equation can be written

$$\rho_2 u_2^2 A_2 - \rho_1 u_1^2 A_1 = (P_1 - p_2)[\alpha A_2 + (1 - \alpha) A_1] \quad (8.76)$$

and with the aid of the continuity equation solving for $u_2$ gives

$$u_2 = u_1 \left\{ 1 + \left( \frac{P_1 - p_2}{\rho_1 u_1^2} \right) \left[ \frac{\alpha A_2}{A_1} + (1 - \alpha) \right] \right\}. \quad (8.77)$$

Setting $\alpha = 0$ in the above equation yields the expression used by the jet utility program for the velocity at position 2. Normal shock relations, conservation of mass, ideal gas equation of state, and equality of $A_2$ and $A_3$ are used to relate states 2 and 3:

$$M_2^2 = \frac{M_2^2}{\gamma - 1} \frac{\frac{2\gamma}{\gamma + 1} M_2^2 - 1}{\gamma RT_3} \quad (8.78)$$

$$p_3 = p_2 \left( \frac{2\gamma}{\gamma + 1} M_2^2 - \frac{\gamma - 1}{\lambda + 1} \right) \quad (8.79)$$

$$T_3 = T_2 \left( \frac{1 + \frac{\gamma - 1}{2} M_2^2}{1 + \frac{\gamma - 1}{2} M_3^2} \right). \quad (8.80)$$

The adiabatic energy equation between stations 1 and 2 gives

$$T_2 = T_1 + \frac{(\nu_1^2 - \nu_2^2)}{2c_p}. \quad (8.81)$$

From the equation of state and conservation of mass, $A_2$ can be determined, giving the Mach disk diameter for the Jet model. See (Birch, 1984) and (Birch, 1987) for further details.

The Mach disk location is the distance between exit plane at vessel wall and Mach disk, see figure above. According to Franquet, Perrier, Gibout, Brue (2015), various studies report that the Mach disk location
appears to be mainly governed by the pressure ratio. It is increased by the exit Mach number (for supersonic flow obtained with convergent-divergent nozzles), and it is independent of the fluid.

In practice, a simplifying assumption is made when using the jet utility program and the pseudo-source approach: the distance from release origin to the pseudo-source “position” is neglected. That is, in FLACS-CFD the pseudo-source leak obtained using the jet utility program, is located at the position where the release occurs, with no downstream shift to position 3 in figure above. In general, this simplification is OK for flacs simulations. Formulas estimating the Mach disk location can be found in Franquet, Perrier, Gibout, Bruel (2015).

8.10.2 Ewan–Moodie pseudo-source model

Ewan and Moodie created a different pseudo-source model, which can, for example, be found in the review by Franquet (2015) (cf. the paragraphs “Sonic jet approach” and “Underexpanded jet theory”). The model is based on the following assumptions:

- the temperature at the pseudo-source equals the temperature at the nozzle,
- the velocity at pseudo-source is sonic.

Using the same notation as in the previous subsection, these assumptions lead to the following expressions:

\[ T_3 = T_1, \quad (8.82) \]
\[ v_3 = c_3, \quad (8.83) \]
\[ d_3 = \sqrt{\frac{\rho_1 v_1}{\rho_3 v_3}} \quad \text{(continuity),} \quad (8.84) \]
\[ \rho = f(T, p, M) \quad \text{(equation of state),} \quad (8.85) \]

where \( p, T, v, d, \rho, c \) are pressure, temperature, velocity, module diameter, density, and speed of sound, respectively, and the subscripts \( \infty, 3, 1 \) stand for ambient, pseudo-source, and nozzle.

By introducing the ideal gas equation of state and assuming isentropic flow from the reservoir to the nozzle and from the nozzle to the pseudo-source, it is possible to rewrite the equations above as a function of the known temperature and pressure in the reservoir \( T_0 \) and \( p_0 \):

\[ T_3 = \frac{2T_0}{\gamma + 1}, \quad (8.86) \]
\[ v_3 = \sqrt{\frac{\gamma p_\infty}{\rho_3}}, \quad (8.87) \]
\[ d_3 = \sqrt{\left( \frac{2}{\gamma + 1} \right) \left( \frac{\gamma}{\gamma - 1} \right) \frac{p_0}{p_\infty}}, \quad (8.88) \]
\[ \rho_3 = \frac{p_\infty M}{RT_3}, \quad (8.89) \]

where \( \gamma \) is the adiabatic index (heat capacity ratio), \( R \) is the universal gas constant, and \( M \) is the molar mass of the released gas.

8.11 Leaks

Gas leaks occur frequently in offshore and onshore installations and elsewhere, either during production, processing or transport of the gas. Hydrocarbons and other highly flammable gases are used widely throughout the industry and may be a source of potentially dangerous gas explosions. In addition, spills of flammable liquids may cause accumulation of explosive gas clouds under certain conditions. Experiments have clearly shown that even small amounts of gas mixed with air at the right concentration can result in
very strong explosions. By using the dispersion and explosion simulation capabilities of FLACS-CFD, it is possible to analyse realistic scenarios where a single gas leak develops into an explosive gas cloud, resulting in several potential gas explosion scenarios, which can be simulated and studied. One primary interest may be to vary the leakage itself, changing the flow rate, location, and duration. Other sensitivities may also be studied such as modifying the ventilation and wind conditions for a given geometrical layout. Finally, the variation of ignition time and location may be made during the explosion studies.

There are two ways to represent leaks in FLACS-CFD: point leaks (each leak is contained inside a single grid cell), and area leaks (leaks covering more than one grid cell, with a rectangular or elliptic shape and one of a range of spatial source density distributions).

The release model for leaks in FLACS-CFD ensures that the desired mass flow or velocity of gas is set at the control volume where the leak is located. Other properties, such as the temperature and relative turbulence intensity and length scale for the leak, must be specified. The figure below shows how a jet is represented on the grid:

![Figure 8.3: Leak definition](image)

It is possible to specify one or more leakage points in a FLACS-CFD simulation. The leakage is intended to model the flow of gas from a reservoir into the simulation volume.

The properties of a leak are prescribed in the scenario file (position and size of the leaks) and in a leak file, which contains time-varying values for leak parameters. The format of these files is fully compatible with other leak types. The parameters that describe a leak are:

```plaintext
LEAKS
INSERT 1
TYPE "JET"
POSITION 1.0, 1.0, 1.0 (m)
OPEN_SIDES "+X"
START_TIME 1.0 (s)
DURATION 10.0 (s)
OUTLET VESSEL
```

These fields can be set in both the scenario file and the leak file using CASD and are described in the leaks section of the CASD chapter.

See also:

Leak files are described in Input files for FLACS-CFD simulations and may be created manually or by using the jet utility program, which is implemented in the Leak Wizard in CASD.
8.11 Leaks

8.11.1 Point leaks

Point leaks are contained within a single control volume (grid cell). The area porosity of the open control volume face is adjusted to ensure that the flow rate is correct. If the grid spacing is large compared to the size of jet orifice, a good representation of the detailed flow in the near-orifice region is not possible. This may lead to excessive smearing of the gradients of the flow. In such cases (which may be the normal practical case), one must realize that the simulated gas concentration is locally underestimated, although the total amount is correct. The grid resolution may be especially important for jets impinging on nearby walls and jets with a strong cross-wind influence. It is also important that a point leak is not located inside a grid cell that also contains a solid wall because the wall may be moved during the porosity calculations (walls are adjusted to grid lines), and a leak that was originally located on one side of it may then be on the other side of it, similarly to the considerations for monitor points.

Note:

The Leak Wizard can be used to set up point leaks in CASD.

8.11.2 Area leaks

Leaks that cover more than one grid cell are treated as area leaks, and may have non-constant properties across the release area.

If a FLACS-CFD run uses a symmetry plane that intersects a leak, then the leak position and size should be such that the symmetry plane exactly halves the leak box. All extensive quantities given in the leak file still refer to the complete leak area, so that the leak file is usable for symmetric and normal runs without changes. The use of symmetry planes is generally not recommended with FLACS-CFD as it may prevent some (unsymmetrical) three-dimensional flow structures from developing in a realistic manner.

Area leaks in FLACS-CFD may have properties like

- time-varying area $A$, leak rate $\dot{m}$, etc.,
- spatially varying mass flux according to given profiles (uniform $k$, $\epsilon$ and $T$).

and must have one of the following shapes:

- rectangular area aligned with the grid, or
- elliptic area where the main axes are aligned with the grid.

By reducing the area porosities for grid cells close to/in the leak, the shape can be modified to obtain leaks that are not elliptic or rectangular. Time-varying areas allow the leak to grow and shrink during the leak duration, with the shape being kept at a fixed aspect ratio. Each area leak is bounded by a rectangular set of grid cells that contains the complete leak area with its maximum size during the simulation time. This leak box is specified per leak in the scenario file with its position $(x, y, z)$ and size $(s_x, s_y, s_z)$ given in Cartesian coordinates. Within the leak boxes, leak properties are computed and the boxes should therefore be chosen as small as the given case allows. However, the leak box 'snaps' to the nearest grid lines, which can lead to the leak being smaller than expected if the position and size result in a boundary that is not on grid planes. Therefore it may be necessary to specify the leak box slightly bigger than the theoretical values based merely on the leak area; increasing the leak box size to an odd number of grid cells may also help in case of area leaks that have a strongly time-dependent size and reduce to a very small area at some point: in this way it is guaranteed that at least one cell center will always be inside the area leak.

In the specification of the leak box size, one dimension must have a zero extent, while the other two have to span at least two cells. It is recommended not to apply any grid stretching within a leak box, so that all leak cells are of equal size.

For notational simplicity, in the rest of this section a single elliptic leak in the x-y-plane will be assumed (i.e., $s_z = 0$). The centre of the leak is then given as

$$ (c_x, c_y) = (x + \frac{s_x}{2}, y + \frac{s_y}{2}) . \quad (8.90) $$
Since \( s_x \) and \( s_y \) need not be equal, elliptic leaks with an aspect ratio \( a = s_x/s_y \) can be realized. For a given leak, the leak file specifies the leak area \( A \) as a function of time within the simulation interval. Based on the area value, the leak may be smaller than the maximum size ellipsis that would fit into the rectangular area of the leak box, as illustrated in the figure below.

![Leak box and size parameters.](image)

Since the aspect ratio \( a \) of the leak is fixed for the leak box, the area \( A \) determines the radii of the elliptic release:

\[
A = \pi r_x r_y = \pi a r_y^2 \Rightarrow A = \frac{A}{\pi a}, \ r_x = a r_y.
\]  

(8.91)

Notably, a complete ellipse is considered when determining the radii from the area; if parts of the surface are obstructed (area porosity less than one) then the leak flux density will be adapted on the remaining open part of the ellipse to match the given rate. There must be at least one (open) grid cell in the leak; if this is not the case, Flacs may open one or several cells for the leak at the center of the leak box.

In case of working with a Gaussian function, the above radii are translated to the standard deviations by setting:

\[
\sigma_x = \frac{2}{5} r_x, \ \sigma_y = \frac{2}{5} r_y
\]  

(8.92)

The release flux density across the leak area can be described by the following profile functions:

- A Gaussian function with standard deviations \( \sigma_x \), and \( \sigma_y \) in the \( x \) and \( y \) direction, respectively:

  \[
f_G = \exp \left[ - \left( \frac{(x-c_x)^2}{2\sigma_x^2} + \frac{(y-c_y)^2}{2\sigma_y^2} \right) \right]
\]  

(8.93)

- A step function with given radius and unit height:

  \[
f_H = \begin{cases} 
  1 & \text{if } \frac{(x-c_x)^2}{r_x^2} + \frac{(y-c_y)^2}{r_y^2} < 1, \\
  0 & \text{otherwise.}
\end{cases}
\]  

(8.94)

- A parabolic function (resembling laminar pipe flow) with given radius and unit height:

  \[
f_P = \max \left( 1 - \left( \frac{x-c_x}{r_x} \right)^2 - \left( \frac{y-c_y}{r_y} \right)^2, 0 \right).
\]  

(8.95)
For rectangular leaks, corresponding versions for the step function and the parabola have been implemented, see also the figure below.

<table>
<thead>
<tr>
<th>shape = Elliptic</th>
<th>shape = Rectangular</th>
</tr>
</thead>
<tbody>
<tr>
<td>profile = Gaussian</td>
<td>N/A</td>
</tr>
<tr>
<td>profile = Uniform</td>
<td></td>
</tr>
<tr>
<td>profile = Parabolic</td>
<td></td>
</tr>
</tbody>
</table>

Figure 8.5: Profile functions for the area leak flux density on elliptic and rectangular area leaks.

To obtain the flux density across the leak area, the profile function is scaled by a factor $M$ (the maximum flux density) as follows: First the discrete integral of the profile function $f$ over the leak area is computed, weighed by the area porosity $\beta_i$ perpendicular to the leak plane. Second, the maximum flux density $M$ is computed from the flow rate $\dot{m}$ given in the leak file according to the relation:

$$\dot{m} = M \int_A f(x, y) \beta_i(x, y) dA.$$  

Finally, the flux density per grid cell is given by the chosen profile function scaled by the factor $M$. Depending on the type of the leak (jet or diffusive), the flux density is, respectively,

- inserted as a source in each individual cell, which is closed off on all boundaries except the one in the leak direction, or
- applied as a source in the cells at or above the leak plane (which should not lie exactly in a coordinate plane, see examples in later sections) and no faces are artificially closed off.
When it is required to obtain gas properties (like density) from the flux, the temperature given in the leak file is used. The turbulence quantities are also taken from there and prescribed as constant over the entire leak area.

Note that the model for area leaks in the core simulator flacs, uses the 'best fit' number of Control Volumes (CVs of the numerical grid) within the area leak box for each time step, depending on the leak area defined in the cl-file (and this leak area will in general vary as function of time). Each CV within the area leak box, is in general either totally open, or totally closed, with respect to release of gas, for a specific time instance, but which CVs that are totally open/closed will in general vary as function of time. The given, time-dependent, mass rate \([kg/s]\) of the release is in general conserved, but when the 'best fit' number of CVs within the area leak box, changes from one time step to the next time step (e.g. when the leak area defined in the cl-file is shrinking), then a kind of discontinuity ('jump') with respect to velocity, might be seen in the detailed flow results. A more refined grid (compared to a more coarse grid), covering the largest leak area for your leak profile, is expected to reduce the magnitude of such velocity 'jumps'.

### 8.11.2.1 Examples of area leaks

![Flux density](image)

Figure 8.6: Normalised flux density for an elliptic leak with a Gaussian profile function on a stretched grid. Two parts of the domain have been (partially) obstructed (50 \% porosity for the lower left box, 0 \% for the upper right one); the flux densities over the whole domain are scaled to match the mass flux specified in the leak file.
8.12 Entrainment model

Note:
To verify which cells actually contribute to an area leak, given the specified leak box and size, it is useful to check the results in Flowvis, plotting the velocities just downstream of the leak with interpolation disabled.

8.11.3 Time-varying leaks

Time-varying leaks can result from depressurization of a reservoir. For these cases, the expanded leak area varies as the mass flow rate changes. When the variability of the expanded leak area is high, the leak should be represented as an area leak, rather than as a point leak. Each grid cell comprising an area leak has porosity equal to one (fully open cells), and the number of cells that comprise the leak area can be set to vary with time, as determined by the evolution of the expanded leak area, which follows the evolution of the mass flow rate.

It is not recommended to represent time-varying leaks as point leaks because FLACS-CFD sets the porosity of the outflow cell face for a point leak so that the effective cell area is equal to the expanded leak area. When the porosity is less than 50%, the effective cell size is too coarse to represent the outflow from the leak, and simulation results are degraded.

8.11.4 Potential issues with oblique jet leaks

The Cartesian grid used in FLACS is most optimal for axis aligned jet leaks; both in terms of accuracy and calculation speed. The simulation of (oblique) jets that are not aligned with any of the coordinate directions of the Cartesian grid is challenging and may give poor results. Such oblique jets can lead to very diffuse flow, which is often bifurcated. In addition, only very limited validation has been done for oblique jets. When practical it therefore recommended to avoid using oblique jets, by rotating the entire scenario (incl. geometry), so the leak can be modelled as axis aligned. In cases where this is not practical (e.g. when the negative effect of rotating the geometry is expected to be stronger), the following approach is recommended:

1. Use the leak outlet: direction cosine vector to define the direction of an oblique jet. If possible, model at an angle of 45° which seems to give better results than other oblique angles.
2. Use cubical cells for leak refinement within recommend aspect ratio.
3. Use as finer cubical cells as possible along the flow direction within the core domain.
4. Oblique jets are computationally intense and take large time to complete so decrease domain size to keep the cell count low (i.e. less than 250k).
5. To verify oblique jet setup gives acceptable results, first run the oblique jet without geometry and compare results (e.g. LFL cloud width and length) to results of the same jet when it is modelled as axis aligned.

8.12 Entrainment model

The entrainment model describes the near-source development of a turbulent jet and is based on the following assumptions:

1. the jet is dominated by momentum (so called pure jet) so that the entrainment is determined by the stream-wise shear;
2. the surrounding ambient fluid is quiescent;
3. the radial velocity, concentration and temperature profiles are uniform (so called top-hat profile);
4. the jet flow is steady-state;
5. the jet is axisymmetric (i.e. the jet is released from an axisymmetric nozzle);
6. turbulent fluxes of scalars and momentum are proportional to the mean values, so there is no need for a separate representation of those terms in the model;
7. the mass entrainment is proportional to the jet velocity.

Assumptions 1 and 2 reflect the purpose of the model, which is to describe the near field region where the jet velocity is high compared to the ambient velocity. Assumption 3 allows simple integration of quantities over the jet cross-section and is consistent with the sub-grid application of the model. Assumptions 4 to 6 are in agreement with the typical release from pressurized conditions and assumption 7 is the classic entrainment hypothesis which replaces the description of the turbulent processes in the integral formulation. In the remainder of this section, the term air is used for the ambient fluid regardless of its composition.

### Table 8.3: Notation for quantities in the entrainment model.

<table>
<thead>
<tr>
<th>symbol</th>
<th>explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_a, f_c )</td>
<td>air and fuel volume fraction</td>
</tr>
<tr>
<td>( \rho_a, \rho_c )</td>
<td>air and fuel density</td>
</tr>
<tr>
<td>( c_a, c_c )</td>
<td>air and fuel specific heat capacity</td>
</tr>
<tr>
<td>( T_a )</td>
<td>air temperature</td>
</tr>
<tr>
<td>( \rho )</td>
<td>jet density</td>
</tr>
<tr>
<td>( T )</td>
<td>jet temperature</td>
</tr>
<tr>
<td>( A )</td>
<td>jet area</td>
</tr>
<tr>
<td>( u )</td>
<td>jet velocity</td>
</tr>
<tr>
<td>( E )</td>
<td>entrainment rate</td>
</tr>
</tbody>
</table>

The mass, momentum and enthalpy balance equations, plus the entrainment rate formulation (Ricou & Spalding, 1961), can be written as

\[
\frac{\partial}{\partial s} f_a \rho_a u A = E \quad \text{air mass balance (8.97)}
\]

\[
\frac{\partial}{\partial s} f_c \rho_c u A = 0 \quad \text{fuel mass conservation (8.98)}
\]

\[
\frac{\partial}{\partial s} \rho u^2 A = 0 \quad \text{momentum conservation (8.99)}
\]

\[
\frac{\partial}{\partial s} (f_a \rho_a c_a + f_c \rho_c c_c) T u A = E c_a T_a \quad \text{enthalpy balance (8.100)}
\]

\[
E = 2\sqrt{\pi} 0.0806 \rho_a u \left( \frac{A}{\rho_a} \right)^{\frac{1}{2}} \quad \text{entrainment rate (8.101)}
\]

and completed by the conservation of the mixture fractions and the ideal gas equation of state constitute the jet entrainment model.

### 8.13 Jet release of liquefied gas - FLASH utility program

Storage of liquefied gas under high pressure at temperatures above the normal boiling point of the gas is common in the industry. In this context, the gas is a substance that may be in either gaseous phase or liquid phase depending on the thermodynamic conditions, but which at atmospheric pressure and normal temperatures is in gaseous phase, e.g., ammonia or propane.

When a jet release of liquefied gas into the atmosphere occurs, some or all of the liquefied gas will evaporate quickly, or in other words flash off. This is due to the fact that the storage temperature is higher than the normal boiling point of the gas, and consequently the storage pressure is higher than the atmospheric.
pressure (assuming that there is thermal equilibrium inside the storage tank the storage pressure is the vapour pressure at the temperature inside the tank). The flashing process is basically determined by how much energy the liquefied gas can draw from itself in order to evaporate. The fraction of the liquefied gas that flashes off depends on the storage temperature, the normal boiling point, the heat capacity of the liquefied gas, and the latent heat of evaporation.

Immediately after the flashing, little air has been mixed with the released gas, and the temperature of the gas both in gaseous phase and liquid phase is approximately the normal boiling point of the gas. The released gas continues downwind as a two-phase flow. Some of the liquefied gas may rain out and form a pool on the ground, the rest will remain in the jet as an aerosol or spray, and will gradually evaporate. After the flashing process, air is gradually entrained into the jet. When air is mixed with released gas the partial pressure of the released gas in gaseous phase is less than the atmospheric pressure and even more liquefied gas can evaporate. Generally, the cooling effect due to evaporation is so large compared to the heat flow of the relatively warm entrained air that the temperature of the jet decreases below the normal boiling point of the released gas. The gradual evaporation of the liquefied gas continues downstream of the jet until the liquefied gas has completely evaporated. Then the temperature of the jet increases as more air is entrained into the jet. Eventually the jet dies out downstream and a vapour cloud region is formed, as illustrated by the different regions in the sketch describing the Flash utility.

Presently, the FLACS-CFD code simulates only gaseous flow and two-phase flow of an aerosol is not modelled directly. The question then arises; with a jet release of liquefied gas as described above, how can this be included in a FLACS-CFD simulation. One possibility is to try to model the jet of liquefied gas with a jet of only vapour of released gas, the temperature being equal the normal boiling point (knowing that in general the temperature of the released gas after flashing is the normal boiling point), and with the same mass and momentum flow as through the orifice. However, our experience has been that when doing this, the released gas spreads out from the jet axis in the numerical calculations much less than it should. The most important reason for this is expected to be the phase transition from liquid to gaseous phase, including the acceleration of the jet as a result of the depressurisation (flashing) when the jet exits the orifice. Hence, this phase transition should be modelled, and this has been done. It is assumed that the jet axis lies in the horizontal plane and that the jet may expand freely in the region of interest.

A jet release of liquefied gas under high pressure involves several processes that are modelled using conservation laws and empirical relations. When the liquid exits the orifice, some of it flashes off. The mass fraction of vapour after this rapid evaporation is an important parameter. After the flashing, the flow of the jet continues downstream as a two-phase flow (aerosol). Knowledge about the droplet distribution is needed to estimate the mass fraction of liquid that rains out and forms a pool on the ground. The fraction of liquefied gas that remains in the jet will eventually evaporate completely as more and more ambient air is entrained into the jet. It is important to be able to characterise the jet as function of the axial distance. This includes the mass flow, the momentum flow, the diameter of the jet, the mass fraction of released gas, and the thermodynamic condition. A detailed description of how this has been done is given in (Salvesen, 1995).

The implemented model is not an integrated part of the FLACS-CFD code, but a separate utility program called FLASH. This program gives as output data that can be used as input to the FLACS-CFD code. To be more specific; the program calculates the mass flow of released liquefied gas that rains out and forms a pool on the ground, the rest of the released gas evaporates inside the jet. The axial distance when all the liquefied gas inside the jet has just evaporated, is calculated. At this axial distance, the jet is characterised by the jet diameter, the mass flow rate, the temperature, the mass fraction of released gas in the entrained jet (from this fraction the corresponding equivalence ratio of the mixture of released gas and air, can be calculated). Other parameters can also be deduced (e.g. the density of the jet, the volume fraction of released gas in the entrained jet).

In practice, when using the FLACS-CFD code, a jet release is included in the scenario at the axial distance where all the liquefied gas inside the jet has just evaporated. Data from the utility program FLASH is then used as input to FLACS-CFD to characterise the jet at this axial distance. The FLACS-CFD code will model the jet further downstream. Further downstream, the jet contains the released substance only in gaseous phase.

Eight types of released gas are possible to use in the present version of the utility program FLASH. These are listed below (in alphabetical order). Of course, in the future it is possible to extend this list. Butane, as used in the model, is assumed to be n-butane and not isobutane (these two types of butane have somewhat
The model implemented in the utility program is valid within a certain range of applicability. It is assumed that only liquefied gas exits the orifice (not a mixture of liquid and gas in gaseous phase). The temperature of the liquefied gas before it exits the orifice is assumed to be larger than the normal boiling point, but not larger than the critical temperature. The reason for this assumption is that if the temperature of a gas in thermodynamic equilibrium is larger than the critical temperature, the gas will be in gaseous phase at any pressure, and there cannot exist any liquefied gas. And if the temperature of a gas in thermodynamic

The orifice pressure $P_0$ [Pa] (also called driving pressure) is, if not otherwise specified, assumed to be related to the ambient atmospheric pressure $P_{\text{atm}}$ [Pa], the density of liquefied gas $\rho_{\text{liq}}$ [kg/m$^3$], the orifice area $A$ [m$^2$], the discharge coefficient $C_{\text{dis}}$ [-], and the mass flow rate $m_0$ [kg/s], by (cf. the theory in (Lees, 1980))

\[
m_0 = AC_{\text{dis}}\sqrt{2\rho_{\text{liq}}(P_0 - P_{\text{atm}})}.
\] (8.102)

The discharge coefficient is approximately 0.62 for a sharp-edged orifice (this is the default value in the utility program FLASH), and 1 for a well rounded nozzle on a vessel. More detailed information about discharge coefficients can be found e.g. in (Coulson & Richardson, 1977).

There are two options when the driving pressure $P_0$ and the mass flow rate $m_0$ are determined. The first option is the following: The mass flow rate is calculated from the driving pressure and the relation given above. The driving pressure is assumed to be the sum of three terms; the vapour pressure of the liquefied gas at the orifice temperature, the hydrostatic pressure of the liquefied gas (determined by the height of liquid from the level where the release is and up to the level being the interface between liquid and gas), plus the partial pressure of other gas(es) (gaseous phase), if there is any, inside the storage tank (an other gas may e.g. be nitrogen having a low critical temperature, 126.2K). In the utility program FLASH, one is asked to prescribe the height of liquefied gas mass (if this height is set to zero, the hydrostatic pressure does not contribute to the driving pressure), and to set the partial pressure [kPa] of other gas(es) inside the storage tank; zero pressure implies no contribution to the driving pressure from other gases than the released gas itself.

The second option is the following: The orifice pressure $P_0$ is either specified directly for FLASH, or it is calculated from the mass flow rate $m_0$, and the relation given above. In the cases where both the orifice pressure $P_0$ and the mass flow rate $m_0$ are specified, it is not in general expected that the above relation is satisfied for the assumed value of the discharge coefficient. Actually, in this case the above relation is valid with a new value of the discharge coefficient defined by this relation (it is easy to calculate this value by hand, too).

If the orifice pressure exceeds the vapour pressure of the liquefied gas at the orifice temperature, the exceeding pressure is assumed to be due to the hydrostatic pressure of the liquefied gas, plus the partial pressure of other gas(es) (gaseous phase), if there is any, inside the storage tank. If the orifice pressure is less than the vapour pressure of the liquefied gas at the orifice temperature, this is assumed to be caused by a pressure loss from the tank to the orifice (drag forces, cooling of the liquefied gas giving a lower vapour pressure at the reduced temperature). However, the model is expected to be less accurate if the orifice pressure is very low compared to the vapour pressure, and one may also ask whether the scenario is physically realistic with such a low orifice pressure compared to the vapour pressure. In general it is advisable to choose a scenario where the orifice pressure is larger or equal the vapour pressure of the liquefied gas at the orifice temperature. If the orifice pressure $P_0$ calculated by the FLASH program is low compared to the vapour pressure of the liquefied gas at the orifice temperature, you may consider to run the FLASH program once more for a reduced orifice area (with all other parameters fixed this will give an increased orifice pressure), or to specify directly an increased orifice pressure. Actually, using the relation above, it is easy to calculate, for a fixed mass flow rate, which value of the orifice area that corresponds to an orifice pressure equal the vapour pressure of the liquefied gas at the orifice temperature (values of the density of liquefied gas $\rho_{\text{liq}}$, and the vapour pressure of the liquefied gas at the orifice temperature can be obtained as output from the utility program FLASH).

The model implemented in the utility program is valid within a certain range of applicability. It is assumed that only liquefied gas exits the orifice (not a mixture of liquid and gas in gaseous phase). The temperature of the liquefied gas before it exits the orifice is assumed to be larger than the normal boiling point, but not larger than the critical temperature. The reason for this assumption is that if the temperature of a gas in thermodynamic equilibrium is larger than the critical temperature, the gas will be in gaseous phase at any pressure, and there cannot exist any liquefied gas. And if the temperature of a gas in thermodynamic
equilibrium is equal or less than the normal boiling point, then the vapour pressure is equal or less than one atmosphere (standard atmospheric pressure, 101.325kPa, approximately equal air pressure at sea level). The model is concerned with liquefied gas under high pressure, i.e. a pressure higher than atmospheric pressure. To ensure that the orifice pressure is larger than the atmospheric pressure, it is assumed that the temperature of the liquefied gas before it exits the orifice is larger than the normal boiling point (in addition to the vapour pressure of the released gas, the hydrostatic pressure of liquefied gas at the level of release, and partial pressure of other gas(es) inside the storage tank, may also contribute to the orifice pressure). The critical temperatures and the normal boiling points for the gas types allowed in the code are:

Table 8.4: Allowed pre-defined gas types (in alphabetical order) in the utility program FLASH.

<table>
<thead>
<tr>
<th>Gas type</th>
<th>Critical temperature [K]</th>
<th>Normal boiling point [K]</th>
</tr>
</thead>
<tbody>
<tr>
<td>acetylene</td>
<td>308.3</td>
<td>189.2</td>
</tr>
<tr>
<td>ammonia</td>
<td>405.6</td>
<td>239.7</td>
</tr>
<tr>
<td>butane</td>
<td>425.2</td>
<td>272.7</td>
</tr>
<tr>
<td>chlorine</td>
<td>417.0</td>
<td>238.7</td>
</tr>
<tr>
<td>ethane</td>
<td>305.4</td>
<td>184.5</td>
</tr>
<tr>
<td>ethylene</td>
<td>282.4</td>
<td>169.4</td>
</tr>
<tr>
<td>fluoroethylene</td>
<td>327.8</td>
<td>200.95</td>
</tr>
<tr>
<td>hexane</td>
<td>507.4</td>
<td>341.95</td>
</tr>
<tr>
<td>methane</td>
<td>190.6</td>
<td>111.7</td>
</tr>
<tr>
<td>propane</td>
<td>369.8</td>
<td>231.1</td>
</tr>
<tr>
<td>propylene</td>
<td>365.0</td>
<td>225.4</td>
</tr>
</tbody>
</table>

Notably, 273.15 K (0°C) is between the critical temperature and the normal boiling point for all listed gases, except for methane.

The temperature of the ambient air is assumed to be larger than the normal boiling point of the released gas. This ensures that when air is entrained into the jet (when there is still some liquefied gas in the jet that has not evaporated), it gives a positive contribution to the total enthalpy of the entrained jet (mixture of released gas and entrained air).

The model should be used with great care in the case of acetylene. The normal freezing point for acetylene is 192.4K, which is higher than the normal boiling point, 189.2K. The validity of the model is uncertain in this case, but used with care it is expected that the model is better than no model at all. Another aspect concerning acetylene, is that in the industry it is often stored not as a pure substance, but dissolved in acetone.

Vapour pressure as function of temperature is represented by a vapour-pressure equation (the so-called Riedel-Planck-Miller vapour-pressure equation) valid within a certain temperature range. The upper limit is the critical temperature, so it is not possible to exceed this limit the way the vapour-pressure equation is used in the utility program. But it may happen that the temperature of the jet at the axial distance when all liquefied gas inside the jet has just evaporated, is below the lower limit of the valid temperature range. If this should happen, the utility program gives a message about it.

After the utility program has calculated how much of the liquefied gas that remains after flashing, it continues to calculate the mass fraction of the remaining liquefied gas that rains out from the jet and forms a pool on the ground. If this mass fraction, called the rainout fraction, is large, the validity of the model is more uncertain. In case the rainout fraction is larger than 0.99, the axial distance when the liquefied gas inside the jet has just evaporated is not calculated and only a limited amount of output information is given by the program. The rainout fraction may be large when the temperature of the liquefied gas before it exits the orifice, is close to the normal boiling point of the released gas. In general it is expected that the model for atomisation (breaking up of liquid stream into droplets) and subsequent rainout of liquefied gas, is more reliable when the temperature of the liquefied gas before it exits the orifice is large compared to the normal boiling point, or when the orifice pressure is large compared to the ambient atmospheric pressure.

As mentioned above, it is assumed that the jet axis lies in the horizontal plane and that the jet may expand...
freely in the region of interest. Clearly, if the jet hits some kind of obstruction only a short distance downstream of the orifice, affecting the flow field considerably, the model is not applicable. If the jet is not hindered by any obstructions, but the jet axis deviates much from the horizontal plane, the model for rainout of liquefied gas from the jet onto the ground forming a pool, is not expected to be accurate. However, if the rainout fraction is negligibly small based on the (in this case incorrect) assumption of a horizontal jet axis, the rainout fraction is expected to be negligibly small also in the case when the jet axis deviates from the horizontal plane.

Example using the FLASH utility program: Liquefied propane stored at a temperature of $10^\circ C$ is released through an orifice with radius 0.01m giving a cross-sectional area of the orifice equal to $3.14\times 10^{-4}$. The temperature of the ambient air is $15^\circ C$. The driving pressure at the orifice is the vapour pressure of the liquefied gas at the orifice temperature, plus the hydrostatic pressure of liquefied gas corresponding to a height of 0.5m. These data are given as input to the utility program FLASH, whose calculations yield the following results: The rainout fraction based on the assumption of a horizontal jet axis, is negligibly small, $4.75\times 10^{-13}$. At a distance of 3.56m downstream of the orifice all of the liquefied propane inside the jet has just evaporated. This position is given as input to the FLACS-CFD code (usually via the CASD preprocessor) where a jet release is specified in the following manner: A mixture of propane (gaseous phase) and air with the equivalence ratio 5.27 is released as a jet with an area of 0.64m$^2$. The mass flow rate of the entrained jet is 19.25kg/s, and the temperature of the entrained jet is $-74.63^\circ C$. These data, together with the jet direction assumed to be the same as the jet direction downstream of the orifice in the original description, characterises the jet sufficiently when using the jet release model in the FLACS-CFD code. The utility program FLASH gives additional information to the data mentioned above. These are not strictly necessary as input to the FLACS-CFD code. The output from the FLASH code, together with input from the user (example for propane), are listed below.

```
C:\Program Files\Gexcon\FLACS-CFD_21.3\bin>flash

A model of jet release of liquefied gas under high pressure.
Basic assumptions; Liquefied gas (only liquid phase) under high
pressure is released through an orifice. The jet axis is horizontal
and the jet may expand freely in the region of interest. Further
information and guidance are found in the FLACS-CFD User's Guide.

Type of released gas. Allowed types are (alphabetical order);
acetylene
ammonia
butane
chlorine
ethane
ethylene
hexane
methane
propane
propylene
fluoroethylene
user
Write name of released gas:
propane
Give the cross-sectional area of the orifice [m^2]:
3.14159e-4
Give the temperature of liquefied gas at the orifice [Celsius]:
10
Default value of the discharge coefficient of the orifice is 0.62.
This is an approximate value for a sharp-edged orifice on a vessel.
Do you want to redefine this value? (yes/no): n
The mass flow rate of released gas may be calculated from the orifice
pressure at the orifice, option 1, or it may be specified directly,
option 2.
Specify option (integer value 1 or 2):
1
Give the height [m] of liquefied gas, determining the hydrostatic
```
pressure at the level of release (if this height is set to zero, the hydrostatic pressure does not contribute to the orifice pressure):
0.5

Give the partial pressure [kPa] of other gas(es) inside the storage tank (zero pressure implies no contribution to the stagnation pressure at the orifice from other gases than the released gas itself):
0

The vapor pressure of liquefied gas at orifice-temperature is [kPa]: 630.718252028016
The stagnation pressure at the orifice is [kPa]: 633.572962028016
The calculated mass flow rate of released gas is [kg/s]: 4.84813482733071
Give the temperature of ambient air [Celsius]: 15

Results for released propane

AFTER EVAPORATION
Input data needed in the FLACS-CFD code:
Axial distance where all liquefied gas inside the jet has just evaporated [m]: 4.2594E+00
Jet area [m²]: 918.3943E-03
Jet mass rate [kg/s]: 19.2533E+00
Jet temperature [Celsius]: -74.6352E+00
Equivalence ratio of released gas in entrained jet [-]: 5.2769E+00
Additional parameters of the jet:
Jet density [kg/m³]: 1.9467E+00
Jet velocity [m/s]: 10.7691E+00
Mass fraction of released gas in the jet [-]: 251.8078E-03
Volume fraction of released gas in the jet [-]: 181.0686E-03

AFTER RAINOUT
Jet area [m²]: 13.2842E-03
Jet temperature [Celsius]: -42.0500E+00
Jet mass rate (excluding rainout) [kg/s]: 4.8481E+00
Jet density [kg/m³]: 8.5335E+00
Jet velocity [m/s]: 42.7672E+00
Mass fraction of released substance in vapour phase [-]: 269.5951E-03
Mass fraction of liquefied gas that rains out and forms a pool [-]: 855.0201E-12
Mass rate of liquefied gas that rains out and forms a pool [kg/s]: 4.1453E-09

AFTER FLASHING (before possible rainout)
Jet area [m²]: 13.2842E-03
Jet temperature [Celsius]: -42.0500E+00
Jet mass rate [kg/s]: 4.8481E+00
Jet density [kg/m³]: 8.5335E+00
Jet velocity [m/s]: 42.7672E+00
Mass fraction of released substance in vapour phase [-]: 269.5951E-03
Mean droplet diameter of liquefied gas in aerosol [m]: 172.3642E-06
Critical droplet diameter for rainout [m]: 1.8544E-03

AT VENA CONTRACTA (before possible flashing)
Jet area [m²]: 194.7786E-06
Jet temperature [Celsius]: 10.0000E+00
Jet mass rate [kg/s]: 4.8481E+00
Jet density [kg/m³]: 582.0000E+00
Jet velocity [m/s]: 42.7672E+00

Do you want written out some physical parameters of the released gas, used in the model? (yes/no): y
**Molecular weight [-]** : 44.0970E+00  
**Normal boiling point [K]** : 231.1000E+00  
**Critical temperature [K]** : 369.8000E+00  
**Critical pressure [kPa]** : 4.2455E+03  
**Density of liquefied gas [kg/m³]** : 582.0000E+00  
**Density of released gas (gaseous phase) at normal boiling point and atmospheric pressure [kg/m³]** : 2.3255E+00  
**Specific heat of vapor of released gas at constant pressure at normal boiling point [J/(kg K)]** : 1.3247E+03  
**Specific heat of liquefied gas at constant pressure at normal boiling point [J/(kg K)]** : 2.2042E+03  
**Latent heat of evaporation of released gas [J/kg]** : 425.5555E+03  
**Thermal conductivity of liquefied gas [W/(m K)]** : 107.9258E-03  
**Surface tension between liquefied gas and its vapor [N/m]** : 15.4981E-03  
**Dynamic viscosity of vapor of released gas, at normal boiling point [kg/(m s)]** : 6.3740E-06  
**Vapor pressure at the temperature of liquefied gas at the orifice [kPa]** : 630.7183E+00  

Parameters supplied to the FLASH program are summarised in the list below:

- Type of released gas.
- Cross-sectional area of the orifice.
- Temperature of the liquefied gas at the orifice.
- Discharge coefficient of the orifice; this may be set different from the default value if required.
- Temperature of the ambient air.
- The mass flow rate of the liquefied gas is set directly by the user, or it is calculated from the orifice pressure depending on the hydrostatic pressure of a user-given height of liquefied gas, and the user-given partial pressure of other gas(es) (if present) in the storage tank. When the mass flow rate of liquefied gas is set directly, the orifice pressure may be specified, otherwise it is calculated by the program.

### 8.14 Numerical schemes

FLACS-CFD is a computational fluid dynamics (CFD) code solving the compressible conservation equations on a 3D Cartesian grid using a finite volume method. The conservation equations for mass, momentum, enthalpy, and mass fraction of species, closed by the ideal gas law, are included. The conservation equations can be represented in general as:

$$
\frac{\partial}{\partial t} (\rho \phi) + \frac{\partial}{\partial x_j} (\rho u_j \phi) - \frac{\partial}{\partial x_j} (\rho \Gamma_{\phi} \frac{\partial}{\partial x_j} (\phi)) = S_{\phi}.
$$

The in-house development started around 1980, primarily aimed at simulating the dispersion of flammable gas in process areas, and subsequent explosions of gas-air mixtures. Hjertager (1985, 1986) describes the basic equations used in the FLACS-CFD model, and Hjertager et al. (1988) present the results of explosion experiments to develop and validate FLACS-CFD initially. In the course of more than 25 years of development and evaluation of the FLACS-CFD software, the numerical methods have been steadily modified and revised.

The main application of FLACS-CFD has been to perform explosion and dispersion calculations to help in the improvement of oil and gas platform safety, with initial focus on the North Sea. Significant experimental validation activity has contributed to the wide acceptance of FLACS-CFD as a reliable tool for prediction of natural gas explosions in real process areas offshore and onshore.

The numerical model uses a second order scheme for resolving diffusive fluxes and a second-order κ scheme (hybrid scheme with weighting between 2nd order upwind and 2nd order central difference, with delimiters for some equations) to resolve the convective fluxes.
The time stepping scheme used in FLACS-CFD is a first order backward Euler scheme. Second order schemes in time have been implemented, but are generally not used because they require short time steps. Based on extensive validation, guidelines for time stepping have been established. These rely on CFL-numbers based on speed of sound (CFLC) and flow velocity (CFLV). For explosion calculations CFLC=5 and CFLV=0.5 must be applied (which means that the pressure can propagate 5 cells and the flow 0.5 cells in each time step) to achieve good results. For dispersion calculations, the guidelines are less strict as the results in general depend less on the time steps. In general, it is recommended to apply the default values CFLC=10 and CFLV=1 for dispersion simulations. When the grid is refined near a leak, it is also recommended to consider multiplying CFLC-number with the refinement factor (CFLC-number applied with refined grid, is set to be CFLC-number without refinement multiplied with refinement factor).

The SIMPLE pressure correction algorithm (Patankar, 1980) is applied, and extended to handle compressible flows with additional source terms for the compression work in the enthalpy equation. Iterations are repeated until a mass residual of less than $10^{-4}$ is obtained.

8.15 The frozen cloud concept

The frozen cloud principle assumes that for a given dispersion or ventilation scenario the fuel volume fraction is approximately proportional to the ratio $\dot{m}_1 / \dot{V}_1$. Based on this assumption and a fuel cloud for given leak rate $\dot{m}_1$ and ventilation rate $V_1$, one can produce the fuel cloud for a different leak rate $\dot{m}_2$ and/or ventilation rate $V_2$ by scaling the concentrations from the FLACS-CFD results for $\dot{m}_1$ and $V_1$ with factor $\frac{\dot{m}_2}{\dot{m}_1} \frac{V_1}{V_2}$. This factor is given as `mix` value in the cs.MON file as exemplified below. FLACS-CFD supports the frozen cloud concept by being able to produce output (e.g. Q8, Q9) for several `mix` values in a single simulation. Thereby the approach can, for example, be used in a probabilistic analysis to include a number of scenarios with varying leak rates and/or ventilation rates for estimating event frequencies.

An example of setting up and exploiting the frozen cloud functionality in FLACS-CFD is given below; it makes use of the monitor files (cs.MON and rt.MON).

8.15.1 Example: dispersion and ventilation study

Prepare a FLACS-CFD dispersion/ventilation simulation in the usual way, say for scenario 123456. The input files for the scenario should include the file cs.MON file cs123456.MON.

To produce different gas clouds, cs123456.MON could, for example, have the following contents:

```
VERSION 1.0
# name=<suffix>; output file will be named rt123456.MON.<suffix>
# start=x,y,z; start position of region in space considered, unit [m]
# end=x,y,z; end position of region in space considered, unit [m]
# see=1.0; in general this parameter is set to a fixed value 1.0
# mix=<fraction>; when the output in the file rt123456.MON.<suffix> is
# calculated, the extrapolated fuel concentration in every control
# volume is calculated as the fuel concentration of standard FLACS-CFD
# multiplied with the factor <fraction>
volume(name="123455",start=138,-21,32,end=198,21,48,output="FUEL(see=1.0,mix=0.8)"
volume(name="123456",start=138,-21,32,end=198,21,48,output="FUEL(see=1.0,mix=1)"
volume(name="123457",start=138,-21,32,end=198,21,48,output="FUEL(see=1.0,mix=1.2)"
```

Based on this input, three rt.MON output files (text files) with the names

- rt123456.MON.123455
- rt123456.MON.123456
- rt123456.MON.123457

will be generated and contain the output for gas clouds based on the scenario and scaling factors 0.8, 1.0, and 1.2, respectively, as explained above. Using the frozen cloud concept, it could for example be assumed...
that the output in \texttt{rt123456.MON.123455} is approximately equal to output that would be obtained by a new full FLACS-CFD simulation with a scenario equal to 123456 except that the leak rate is reduced by multiplying with the factor 0.8 (while the ventilation and other parameters are kept fixed).

### 8.15.2 Recommendations

The accuracy of the frozen cloud approach is scenario-dependent. For example, when both the wind speed and fuel release rate are low, then accuracy considerations require a higher number of simulations rather than following the frozen cloud approach. It may also help to estimate the result for a non-simulated configuration by interpolating/extrapolating it from more than one simulation with closely related parameter sets. Extrapolation of the leak rate or wind speed resulting in a factor $\text{mix}$ of less than 0.5 or more than 2 is not recommended.

### 8.16 Pool model

Pool simulations can involve two types of pool: non-spreading and spreading. A non-spreading pool is stationary and does not spread or flow, for example because the pool is confined by a bunding wall. A spreading pool may spread or flow under gravity or pressure forces. When a bunding wall is present, the maximum pool extent is determined by the bunding wall itself, although the pool may not reach this maximum extent. In the case of no bunding wall (for example, a pool that spreads over a water surface), the maximum pool extent depends on flow and evaporation conditions, which may lead the pool growth to reach a steady state (i.e., the pool size may reach a peak and remain there, not growing further).

![Figure 8.7: (a) Non spreading pool (pool growth limited by bunding wall), (b) Spreading pool.](image)

This section presents the governing equations and the expressions for friction, heat and mass transfer for a spreading pool. A spreading pool model is enabled by setting \texttt{STATIC\_POOL} to 0 in the Pool section of the scenario file. The models have been validated for pool spread with and without obstacles at adiabatic conditions, for evaporation of a liquid methane pool at rest on soil and for spreading of LNG on water.

**Attention:**

The Antoine vapour pressure equation for a species is a polynomial fit which is valid for a specific temperature interval. When running a pool simulation in FLACS-CFD, a warning is printed to the \texttt{Simulation log file - tt-file} if the predicted temperature is outside the valid temperature interval for the liquid modelled. Currently this \textbf{temperature interval is fixed for LNG}. Other liquids than LNG may be specified through either the predefined and/or user-defined species, and used in the pool model, but the respective temperature intervals (for the Antoine equation) cannot be specified. Hence, if the predicted temperature is outside the valid interval for the liquid modelled (different from LNG) a warning system is missing.

When considering a pool consisting of a mixture of different fuel substances, FLACS-CFD will model properties, including a possible phase change, as if it occurred for a single “pseudo-fuel” with suitably averaged property values, rather than considering evaporation or other processes per species. This
approach has shown to be appropriate for the abovementioned applications. However, care should be taken when modelling mixtures of fuels with very different properties, especially when there is no dominant component in the mixture or the ambient conditions are close to, for example, the boiling point of at least one of the components. In general, cases that raise concerns regarding the above exceptions are likely outside the validation range of the pool model. A mixture of different fuel substances, for example gasoline, may be analyzed based on the vapor pressure of the pure components, and the vapor pressure of the mixture, by applying the Raoult’s law. The Raoult’s law states that the partial pressure of each component of an ideal mixture of liquids is equal to the vapor pressure of the pure component multiplied by its mole fraction in the mixture. The vapor pressure of the mixture can be obtained by summing the partial pressure of the components (the law assumes that the intermolecular forces between unlike molecules are equal to those between similar molecules).

The evaporation rate of a multi-component pool is initially (not accounting for the aging of the pool) equal to the evaporation rate of a pure component with the same vapour pressure and the same thermal properties. Thus, in a FLACS-CFD simulation, the multi-component pool can be replaced by a single-component pool based on vapor-pressure similarity and adopting a sufficient conservatism, taking into account that higher vapor pressure correspond to higher evaporation rate. Another approach is to use a software that can model the time development of the pool with a mixture of different fuels. Such a tool may for example be the consequence modelling tool FRED from Shell. The output from such simulations will be the development in time of the evaporation rate of the pool, the vapor composition, and the composition in the liquid pool. These results may then be used to compare evaporation rates with using pure components and the pool models in FLACS-CFD, or by defining a time dependent leak.

### 8.16.1 Pool motion

In FLACS-CFD, spreading pool (spill) is described by the shallow-water equations in two-dimensions. The shallow-water equations are solved on a Cartesian grid identical to the $xy$-grid in FLACS-CFD. The equation solved for the spill height is given by:

\[
\frac{\partial h}{\partial t} + \frac{\partial hu_i}{\partial x_i} = \frac{\dot{m}_L - \dot{m}_V}{\rho_l}
\]  

(8.104)

and the momentum equation is written as:

\[
\frac{\partial hu_i}{\partial t} + u_j \frac{\partial hu_i}{\partial x_j} = F_{g,i} + F_{\tau,i},
\]  

(8.105)

where the gravity term is modelled as follows:

\[
F_{g,i} = h g \Delta \frac{\partial (h + z)}{\partial x_i}.
\]  

(8.106)

The elevation of the ground, $z$, has been included such that spill on sloping surfaces and the effects of obstacles and embankments can be calculated. The parameter $\Delta$ equals one for pools on solid surfaces and $\Delta = (1 - \rho_l/\rho_w)$ for pools on water. The shear stress between the pool and the substrate is given by the general formula:

\[
F_{\tau,i} = \frac{1}{8} f_f |u_i|.
\]  

(8.107)

In the laminar regime, the friction factor $f_{f, \text{lam}}$ is given by:

\[
f_{f, \text{lam}} = \frac{64}{4Re_h}.
\]  

(8.108)

For turbulent flows, the friction factor $f_{f, \text{turb}}$ is estimated by Haaland’s approximation to the Moody chart for small relative roughnesses and Manning factors for large relative roughnesses:

\[
f_{f, \text{turb}} = \begin{cases} 
-1.8 \log \left( \frac{1.72}{Re_h} + \left( \frac{\varepsilon}{12h} \right)^{1.11} \right)^{-2} & \text{if } \frac{\varepsilon}{h} < 0.2 \\
0.125 \left( \frac{\varepsilon}{h} \right)^{1/3} & \text{if } \frac{\varepsilon}{h} \geq 0.2,
\end{cases}
\]  

(8.109)
where $\varepsilon_g$ is the ground roughness. The largest of the laminar and the turbulent friction factors is used:

$$f_f = \max (f_f, \text{lam}, f_f, \text{turb}).$$  \hfill (8.110)

The interfacial friction between a pool and water is less than the friction between pool and solid grounds. On water, the ground roughness equals zero and the laminar friction factor is reduced by a factor 4:

$$f_f, \text{lam, water} = \frac{12}{4Re_h}.$$  \hfill (8.111)

The transport equation for specific enthalpy reads:

$$\frac{\partial h \theta}{\partial t} + u_i \frac{\partial h \theta}{\partial x_i} = \frac{\dot{m}_L}{\rho_l} (\theta_L - \theta) + \frac{1}{\rho_l} \left( \dot{q}_c + \dot{q}_{\text{rad}} + \dot{q}_g + \dot{q}_{\text{evap}} \right),$$  \hfill (8.112)

where

- The first term on the right hand side is due to the leak
- $\dot{q}_c$ is convective heat transfer between pool and air
- $\dot{q}_{\text{rad}}$ is radiative heat transfer from the surroundings and the sun
- $\dot{q}_g$ is heat transfer to the pool from the substrate
- $\dot{q}_{\text{evap}}$ is heat loss due to evaporation.

Expressions for different heat and mass transfer mechanisms will be handled in the following subsection.

### 8.16.2 Pool heat and mass transfer

For cryogenic liquids like liquid $\text{H}_2$, liquid $\text{N}_2$, and LNG, the heat transfer is dominated by the heat from the substrate. Heat transfer from solid and rough grounds and for all grounds at non-boiling conditions is approximated by:

$$\dot{q}_g, \text{cond} = \begin{cases} \lambda_g (T_g^\infty - T_p) \sqrt{\frac{\pi \alpha_g}{t - t_{gw}}} & \text{if } t < 4s \\ \lambda_g (T_g^\infty - T_p) \sqrt{\frac{\pi \alpha_g}{t - t_{gw}}} & \text{if } t \geq 4s, \end{cases}$$  \hfill (8.113)

where $\lambda_g$ is the thermal conductivity of the ground, $\alpha_g$ is the thermal diffusivity of the ground, and $t_{gw}$ is the point in time the ground was wetted. Infinite ground is assumed in the derivation of the expressions above and $T_g^\infty$ is the ground temperature at an infinite position that equals the ground temperature before the ground was wetted. Furthermore, the equation above is only valid for conductive heat transfer. Spreading pools will also have a convective contribution to the heat transfer between the pool and the substrate. The convective heat transfer can be expressed as follows:

$$\dot{q}_g, \text{conv} = 0.0133Re_l^{0.69}Pr_l^{0.4} \frac{\lambda_l}{h} (T_s^g - T_p);$$  \hfill (8.114)

where $\lambda_l$ and $Pr_l$ are conductivity and Prandtl number of the pool liquid and $T_s^g$ is the ground temperature at the surface. The total heat transfer for pools on solid and rough grounds and for non-boiling conditions is found by using a cubic blending function:

$$\dot{q}_g = \left( \dot{q}_{g, \text{cond}}^3 + \dot{q}_{g, \text{conv}}^3 \right)^{1/3}.$$  \hfill (8.115)

In the expression for the convective heat transfer, the surface temperature of the substrate is used, which is calculated as follows:

$$T_s^g = T_g^\infty + \frac{\dot{q}_s}{\lambda_g} \sqrt{\frac{\alpha_g (t - t_{gw})}{\pi}}.$$  \hfill (8.116)
On smooth surfaces such as water and metal, the expressions for boiling heat transfer are used. Nucleate boiling is assumed for slight superheats. Slight superheat is defined as the conditions when the surface temperature of the substrate is at least 4 K higher than the boiling point temperature of the pool liquid and the heat transfer is below the critical heat flux. Cooper’s correlation is used to calculate the nucleate boiling heat transfer:

\[ \dot{q}_{n,b} = (55y_{p}^{0.12}(- \log p_{r})^{-0.55}M^{-0.5}(T_g - T_p)^{3} \]  

(8.117)

where the reduced pressure, \( p_{r} = p_{s}/p_{c} \) where \( p_{s} \) is the saturation pressure and \( p_{c} \) is the critical pressure. The nucleate boiling heat transfer replaces the convective heat transfer in the cubing blending function for the effective heat transfer from the ground. The expressions for transition boiling and film boiling heat transfer and for selection of boiling regime follow Conrado & Vesovic (2000) and references therein. The cubing blending rule is also applied to the transition regime. In the film-boiling regime, the blending rule of Melheim et al. (2009) is used:

\[ \dot{q}_{g} = \begin{cases} \dot{q}_{g,\text{film}} & \text{if } \text{Re}_{h} < 15 \\ \frac{1}{2}\dot{q}_{g,\text{film}} + \frac{1}{2}(\dot{q}_{g,\text{film}}(\frac{1500 - \text{Re}_{h}}{1485}) + \dot{q}_{g,\text{conv}}(\frac{\text{Re}_{h} - 15}{1485})) & \text{if } 15 \geq \text{Re}_{h} < 1500 \\ \frac{1}{2}\dot{q}_{g,\text{film}} + \frac{1}{2}\dot{q}_{g,\text{conv}} & \text{if } \text{Re}_{h} \geq 1500, \end{cases} \]

(8.118)

where \( \dot{q}_{g,\text{film}} \) refers to the film boiling heat transfer for a fluid in rest that can be found in Conrado & Vesovic (2000).

Convective heat and mass transfers are based on boundary layer theory and wall functions similar to those for the momentum equation, see Wall functions, are used. The convective heat transfer reads:

\[ \dot{q}_{c} = \frac{\rho_{g}C_{p}^{1/4}k^{1/2}E_{p,q}(T_g - T_p)}{T^+}, \]

(8.119)

where \( T^+ \) is given by a two-layer model:

\[ T^+ = \begin{cases} \frac{P_{y}^{+}E}{P_{x}^{+}+E} & \text{if } y^{+} < E^+ \\ \frac{P_{x}^{+}E}{P_{y}^{+}+E} & \text{if } y^{+} \geq E^+. \end{cases} \]

(8.120)

The expression for the convective mass transfer is similar to that for heat transfer:

\[ \dot{m}_{c} = \frac{\rho_{g}C_{p}^{1/4}k^{1/2}E_{p,q}}{x^+}(x - x_{sat}), \]

(8.121)

where \( x = P_{g}/P_{0} \) and \( x^+ \) is given by:

\[ x^+ = \begin{cases} \frac{Sc^{+}y^{+}}{} & \text{if } y^{+} < E^+ \\ E^+Sc^{+} & \text{if } y^{+} \geq E^+. \end{cases} \]

(8.122)

Both the sun and the surroundings contribute to the radiative heat transfer:

\[ \dot{q}_{rad} = (1 - \omega)\dot{q}_{s} + \varepsilon_{g}\sigma T_{g}^4 - \varepsilon_{p}\sigma T_{p}^4, \]

(8.123)

where \( \omega \) is the Albedo, \( \varepsilon_{g} \) and \( \varepsilon_{p} \) is the emission coefficients of the surrounding gas and the pool liquid. \( \sigma \) is the Stefan-Boltzmann constant.

Two mechanisms contribute to the evaporation rate, the convective mass transfer and boiling:

\[ \dot{m}_{evap} = \dot{m}_{c} + \dot{m}_{boil}. \]

(8.124)

Evaporation due to boiling hinders the pool temperature to rise above the boiling point temperature and is calculated as follows:

\[ \dot{m}_{boil} = \max \left\{ \dot{q}_{g} + \dot{q}_{c} + \dot{q}_{rad} \frac{\Delta h_{fg}}{\Delta h_{fg}} - \dot{m}_{c}, 0 \right\}. \]

(8.125)

Finally, the heat transfer due to evaporation can be determined by:

\[ \dot{q}_{evap} = -(\dot{m}_{c} + \dot{m}_{boil})\Delta h_{fg}. \]

(8.126)
8.16.3 Coupled pool fire physics

When modelling pool fires with FLACS-Fire there are two options: Either one models the evaporation using tabulated burning rates and/or empirical correlations cf. Source term modelling for pool fires, or by modelling evaporation from the pool using a non-spreading or spreading pool model, cf. Pool setup section.

![Figure 8.8: Pool heat transfer](image)

In the coupled pool fire model radiation from the flame is fed back to the pool model through the heat transfer balance for the pool, illustrated in the figure Pool heat transfer. In addition to the heat from the combustion, the heat balance also includes heat from the wind, the subgrade and the heat of evaporation. The radiative heat can be calculated with the Discrete Transfer Model (DTM). A minimum radiative heat on the pool surface may be set to enforce higher evaporation rates prior to the ignition of the combustible cloud forming over the pool, cf. combustion. Combustion modelling currently applies the Eddy Dissipation Concept (EDC). The switches are described in the sections on combustion and pool fire.

8.16.4 Humidity and fog

The pool model of Flacs supports setting the relative humidity in air. The equation of state and specific heat/enthalpy formulation takes into account phase transition of the water vapour. The following is assumed:

- The specific humidity is constant
- The mist follows the gaseous flow
- No condensation on surfaces
- Thermodynamic equilibrium

The relative humidity in percentage is given by:

\[ RH = \frac{P}{P_{sat}(T)} \cdot 100\% , \quad P = P_0 , \quad T = T_0 \]  \hspace{1cm} (8.127)

\[ P : \text{ partial pressure of water vapour [Pa]} \]
\[ P_{sat} : \text{ saturation pressure of water vapour [Pa]} . \]

The saturation pressure of water vapour is given by an Antoine vapour pressure equation:

\[ \ln(P_{sat}) = A - \frac{B}{T + D} . \]  \hspace{1cm} (8.128)

The specific heat for mixture of gas/vapour liquid (mist) is given by:

\[ C_{p,mix} = \frac{dH}{dT} = YC_{p,g} + (1 - Y)C_{p,l} + L \frac{dY}{dT} . \]  \hspace{1cm} (8.129)

The density of the mixture is obtained via:

\[ \frac{1}{\rho_{mix}} = \frac{Y}{\rho_g} + \frac{1 - Y}{\rho_l} . \]  \hspace{1cm} (8.130)
8.16 Pool model

\( P : \) vapour pressure = \( XP_{\text{atm}} \)

\( P_{\text{sat}} : \) vapour saturation pressure = \( X_{\text{sat}}P_{\text{atm}} \)

\( P_{\text{atm}} : \) atmospheric pressure

\( X : \) mole fraction of water vapour

\( X_{\text{sat}} : \) mole fraction of saturated water vapour

\( A, B, C : \) constants for the vapour pressure formula

\( T : \) absolute temperature

\( \rho_{\text{mix}} : \) density of the mixture

\( \rho_g : \) density of the gas/vapour

\( \rho_l : \) density of the liquid

\( C_{p,\text{mix}} : \) specific heat of the mixture

\( C_{p,g} : \) specific heat of the gas/vapour

\( C_{p,l} : \) specific heat of the liquid

\( L : \) enthalpy of evaporation

\( H : \) enthalpy of the mixture

\( Y : \) mass fraction of gas/vapour.

The output variable FOG gives the mass fraction of water in the liquid phase.

Note:

The variable FOG is not always available in the scenario. If not, save and quit CASD and copy the following lines into SINGLE_FIELD_VARIABLES in the file cs000000.dat3:

```plaintext
FOG "FOG" 1 "(-)" N
"Mass fraction of water in the liquid phase"
```

Then start CASD again and open your case.

8.16.5 Grid domains for pool scenarios

Recommendations for configuring the grid for pool scenarios are included in the general recommendations for gridding, however there are different regions of the grid to consider for pool scenarios and this section describes those regions. We consider the horizontal and vertical dimensions separately when defining the regions of the simulation domain that require different resolutions.
8.16.5.1 Horizontal simulation domain (X, Y)

Core Domain
The horizontal core domain should cover the pool, so the extent should equal at least the pool diameter. Under windy conditions, the core domain extent must be increased, depending on the wind velocity. For example, if the wind flow is from -Y to +Y, then then flame will tilt in the +Y direction, and the core domain should therefore be extended in the +Y direction. This creates a secondary core domain. For scenarios with multiple pools, it is recommended to use one large horizontal core domain that covers all pools. If this is not practical, e.g., if the pools are widely separated, then it is important that each pool is contained within a core domain, either on its own or with other pool(s).

Total Domain
The total simulation domain should extend from the pool centre by a distance that is five times the core domain extent. All points of interest should be at least three grid cells away from the domain boundary (to avoid any unphysical boundary effects).

8.16.6 Vertical simulation domain (Z)

The vertical dimension of the simulation domain is divided into three grid zones: the pool region, the core domain and the stretched domain.

Figure 8.9: Core and total domain extents in the horizontal plane.

Figure 8.10: Pool region, core domain and stretched domain.
8.17 Fire model

Pool region
This covers the pool in vertical direction, and should be aligned with any geometry (e.g., any bund or pit that is present). This region starts at the base of the pool (at the substrate-pool interface), and its extent should be such that it includes at least 6 cells above the height of any bund. If there is no bund, then it should extend at least 6 cells above the upper surface of the pool.

Core Domain
This is only required for fire scenarios. The finer resolution provided by the pool region is generally sufficient for evaporation-dispersion scenarios.

\[
CoreDomain_{\text{min}} = PoolRegion_{\text{max}}
\]  

\[
CoreDomain_{\text{max}} = (\text{FlameHeight} - PoolRegion) \times 1.5
\]  

Stretched domain
This is the area of the domain that is outside the area where high resolution calculations are required (i.e., outside the vertical pool region and, for fire, the core domain). Cells in this area may be stretched between the boundary of the higher resolution area and the edge of the simulation domain.

\[
StretchedDomain_{\text{min}} = CoreDomain_{\text{max}} \text{ (for fire)}
\]  

\[
StretchedDomain_{\text{min}} = VerticalPoolRegion_{\text{max}} \text{ (for non-fire)}
\]  

\[
StretchedDomain_{\text{max}} = CoreDomain_{\text{max}} \times 10 \text{ (for fire)}
\]  

\[
StretchedDomain_{\text{max}} = VerticalPoolRegion_{\text{max}} \times 10 \text{ (for non-fire)}
\]  

8.17 Fire model

A release of a flammable gas or liquid will frequently be ignited. The outcome after ignition depends on the fuel type, the amount of fuel, the mixing with the surrounding air, confinement, and the time of ignition. The figure below shows an event tree with the possible scenarios. The worst case scenario is a major explosion. The consequences of a fire are usually less, but fires happen more frequently than
explosions.

Accidental fires represent a significant hazard to people, property and the environment. The risks related to fires are exposure to heat, toxic gases/lack of oxygen, and blockage of escape ways due to smoke/visibility. Exposure to heat is very critical for both human beings and constructions. Materials change their properties when the temperature increases and this can result in serious damage or collapse of entire constructions. For human beings, the tenability limit is approximately 2.5 kW/m² and a heat dose of 2000(kW/m²)⁴/₃s has the probability of fatality of 50% (Mannan, 2005). Carbon monoxide is very toxic and a concentration of 1.28% leads to death within 2-3 minutes (Struttmann, 1998 and Goldstein, 2008).

For a realistic numerical prediction of fire, the complex interaction between turbulent flow, buoyancy, convection, entrainment of air, non-premixed combustion, soot formation, thermal radiation, fluid-structure interaction, dispersion of smoke and toxic combustion products, and effects of mitigating measures needs to be modelled. This section describes the mathematical modelling that is the foundation of the current FLACS-Fire simulator. This includes modelling of non-premixed combustion, soot, and heat transfer by radiation and conduction.

The estimation of the consequences of fire are outside the scope of the current fire modelling, the aim is to provide the necessary input for further analysis, such as wall temperature and wall heat flux, heat flux and heat dose at monitor points, soot levels and estimation of smoke/visibility.

Currently the simulator is limited to single-phase (gas) non-premixed combustion modelling. Compressible and incompressible flow solvers are included. A gas release can be modelled either as a point-leak source or as an area leak. See below for more information about the current models.

Capabilities:

- Compressible and incompressible flow solvers
- Turbulence: standard $k – \varepsilon$ turbulence model
- Combustion: Eddy Dissipation Concept (EDC)
- Radiation model: Hybrid Discrete Transfer Model (DTM) and Far-field radiation model
- Three radiative properties models
- Two soot models: Oxidation-Formation soot model (FOX) and Fixed Conversion rate
- Small-scale, medium-scale and large-scale complex geometries
- Point and area leaks
• Various gases as fuel (more than ten predefined; user-defined) and mixtures of these
• Wind boundary conditions with wind profiles
• Predicting fire growth and behaviour
• Heat transport from fire (both radiation and convection)
• Heat fluxes on surfaces
• Pollutant emissions
• Fire impact on structures and process equipment (Heat fluxes and heat dose)
• Impact of fire (temperature, radiation and smoke) on persons
• 2D and 3D field plots of various fire simulation variables
• Efficient and user-friendly pre-processor (CASD) and postprocessor (FLOWVIS-5)

Application areas:
• both indoor and outdoor fires,
• simulation of jet and pool (modelled as a jet leak) fires,
• offshore installations,
• fires in factory buildings.

FLACS-Fire can be used to simulate
• jet fires in the open,
• jet fires in cross-wind (flares),
• impinging jets,
• jet releases with delayed ignition (dispersion and fire),
• confined jet fires,
• compartment fires (over-, under-ventilated).

Reliable output parameters are
• flame shape,
• flame length,
• gas temperature,
• mixture fraction,
• flame propagation,
• flow (velocities),
• species (O2, CO2, H2O and SOOT),
• radiative and total heat flux,
• heat dose.
8.17.1 Combustion Model

Most fire scenarios involve non-premixed or diffusion flames. For most fires the combustion rate is controlled by the mixing of fuel and air, and simple Mixed Is Burnt (MIB) combustion models can be applied. The current model for the turbulence-chemistry interaction is the Eddy Dissipation Concept (EDC) of Magnussen & Hjertager (1976). In the EDC both fast and detailed chemistry can be applied, and extinction can be modelled.

The transport equation for the fuel mass fraction is written as:

\[
\frac{\partial \rho \tilde{Y}_{\text{fuel}}}{\partial t} + \frac{\partial \rho \tilde{Y}_{\text{fuel}} \tilde{u}_j}{\partial x_j} = \frac{\partial}{\partial x_j} \left( \rho D \frac{\partial \tilde{Y}_{\text{fuel}}}{\partial x_j} \right) - \frac{\partial}{\partial x_j} \left( \rho \tilde{Y}_{\text{fuel}} \tilde{u}_j \right) + \rho \tilde{\omega}_{\text{fuel}}.
\] (8.137)

The turbulent diffusion \( \tilde{Y}_{\text{fuel}} \tilde{u}_j \) and the source term \( \tilde{\omega}_{\text{fuel}} \) have to be modelled. The turbulent diffusion is modelled by using a gradient model (Pope, 2000) and a combustion model models the source term (described below).

8.17.1.1 Eddy dissipation concept

The Eddy Dissipation Concept (EDC) models the interaction between the turbulent flow and the chemical reactions. A physical interpretation of the EDC can be found in (Magnussen & Hjertager, 1976), (Ertesvåg, 2000), here only the main equations and expressions are presented. In the EDC, the source term \( \tilde{\omega}_{\text{fuel}} \) is modelled as:

\[
\tilde{\omega}_{\text{fuel}} = - \frac{\dot{m} \chi_1}{1 - \gamma^* \chi} \tilde{Y}_{\text{min}}
\] (8.138)

\[
\tilde{Y}_{\text{min}} = \min \left( \frac{\tilde{Y}_{\text{fuel}}}{r}, \frac{1}{r} \tilde{Y}_{\text{ox}} \right),
\] (8.139)

where \( r \) is the stoichiometric amount of oxidant.

The parameters in the model are:

\[
\gamma_\lambda = \min \left( 0.8, 2.13 \left( \frac{\nu \tilde{\varepsilon}}{K^2} \right)^{1/4} \right)
\] (8.140)

\[
\gamma^* = \gamma_\lambda^3
\] (8.141)

\[
\dot{m}^* = 2.433 \left( \frac{\tilde{\varepsilon}}{\nu} \right)^{1/2}
\] (8.142)

\[
\dot{m} = \gamma^* \dot{m}^*
\] (8.143)

\[
\chi_1 = \frac{\left( \tilde{Y}_{\text{min}} + \tilde{Y}_{\text{prod}}/(1 + r) \right)^2}{\left( \tilde{Y}_{\text{fuel}} + \tilde{Y}_{\text{prod}}/(1 + r) \right) \left( \tilde{Y}_{\text{ox}}/(1 + r) + \tilde{Y}_{\text{prod}}/(1 + r) \right)}
\] (8.144)

\[
\chi_2 = \min \left( \frac{1}{\gamma_\lambda \tilde{Y}_{\text{prod}}/(1 + r) + \tilde{Y}_{\text{min}}}, 1 \right)
\] (8.145)

\[
\chi_3 = \frac{1}{\gamma_\lambda} \min \left( \frac{\gamma_\lambda \left( \tilde{Y}_{\text{prod}}/(1 + r) + \tilde{Y}_{\text{min}} \right)}{\tilde{Y}_{\text{min}}}, 1 \right)
\] (8.146)

\[
\chi = \chi_1 \chi_2 \chi_3.
\] (8.147)
8.17.2 Radiation model

Thermal radiation plays vital role in overall heat transfer in applications including fire simulations. Thus, in carrying out CFD and other simulations, an accurate modelling of the radiation is required. FLACS-Fire solves the continuity, momentum, and enthalpy equations. Solving the enthalpy equation requires a radiative source term. The general form of energy conservation equation for moving compressible fluid can be given as (Modest, 1993):

$$\rho c_v \frac{DT}{Dt} = \rho c_v \left( \frac{\partial T}{\partial t} + v \cdot \nabla T \right) = \nabla \cdot (k \nabla T) - \nabla \cdot q_R - \rho \nabla \cdot v + \mu \Phi + \dot{Q}'''$$

(8.148)

where \( T \) is the temperature, \( v \) is the velocity vector, \( q_R \) is the radiative heat flux, \( \Phi \) is the dissipation function and \( Q''' \) is the heat generated within the medium.

The governing equation for describing radiation intensity field in an absorbing, emitting and scattering medium is the Radiative Transfer Equation (RTE) (Siegel & Howell, 2001), which is of integro-differential type. The RTE is given by:

$$\mu \frac{dI(\tau, \mu, \phi)}{d\tau} = -I(\tau, \mu, \phi) + (1 - \omega) I_B[T] + \frac{\omega}{4\pi} \int_{\mu' = -1}^{1} \int_{\phi' = 0}^{2\pi} I(\tau, \mu', \phi') \Phi(\mu', \phi'; \mu, \phi) \, d\mu' \, d\phi',$$

(8.149)

where \( \mu \) is the cosine of the polar angle \( \theta \), \( \phi \) is the azimuthal angle, \( I(\tau, \mu, \phi) \) is the intensity along direction \( \mu, \phi \) at optical depth \( \tau \) measured inside the medium perpendicular to the surface, \( I_B \) is the spectral black body intensity at temperature \( T \), \( \omega \) is the single scattering albedo and \( \Phi(\mu', \phi'; \mu, \phi) \) is the scattering phase function.

<table>
<thead>
<tr>
<th>No</th>
<th>Requirements</th>
<th>MCM</th>
<th>DTM</th>
<th>DOM, FVM, DDOM</th>
<th>6-Flux</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Accuracy</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Not accurate</td>
</tr>
<tr>
<td>2</td>
<td>Speed of Computation</td>
<td>Very Slow</td>
<td>Fast</td>
<td>Fast</td>
<td>Very Fast</td>
</tr>
<tr>
<td>3</td>
<td>Treating Complex Geometries</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Poor</td>
<td>Good</td>
</tr>
<tr>
<td>4</td>
<td>Treating Entire Range of Optical Thickness</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Poor</td>
</tr>
<tr>
<td>5</td>
<td>Parallelization</td>
<td>Easy</td>
<td>Easy</td>
<td>Difficult</td>
<td>Difficult</td>
</tr>
<tr>
<td>6</td>
<td>Treating Spectral dependence</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
</tr>
<tr>
<td>7</td>
<td>Treating Non uniform Radiative properties</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Very Good</td>
</tr>
<tr>
<td>8</td>
<td>Treating Anisotropic Scattering</td>
<td>Very Good</td>
<td>No</td>
<td>Very Good</td>
<td>No</td>
</tr>
<tr>
<td>9</td>
<td>Ability to handle conjugate problems</td>
<td>Poor</td>
<td>Very Good</td>
<td>Very Good</td>
<td>Poor</td>
</tr>
</tbody>
</table>

Figure 8.12: Comparison between various radiation models, from very detailed (left) to less detailed (right)

The governing radiative transfer equation is of integro-differential type which makes the analysis difficult and computationally expensive. A literature review shows that several techniques for solving the radiative transfer equation are currently available, for example the two-flux and six-flux methods, the discrete ordinates method (DOM), the finite volume method (FVM), the Monte Carlo method (MCM), etc. These techniques employ advanced mathematics and they generally require extensive programming effort. In addition, each solution methodology needs its specific treatment of the geometry of the system, the boundary conditions and the equilibrium considerations in the medium. The above table presents a comparison between some radiation models (Deiveegan et al., 2006).
For the radiative transfer simulation, several input parameters are needed to characterise the wall and the medium. The parameters that need to be defined are: the gas temperature distribution of the medium, the absorption coefficient distribution of medium (depends on the mole fractions of $CO_2$, $H_2O$, temperature and pressure), temperature and emissivity of the walls, and control parameters like the type of problem (temperature or source specified) and the number of ray directions (Malalasekera et al., 2009).

8.17.2.1 Six-flux radiation model

The six-flux radiation model implemented in FLACS (for Gas explosion) is based on the composite-flux model of Spalding (1980).

The six-flux model solves the RTE by approximating it by six first order differential equations. These differential equations are obtained by discretisation of the angle such that the effects of radiation are accounted for by reference to the positive and negative radiation fluxes in each of the three coordinate directions. Then, using composite-flux definitions, these six equations are converted into three second-order ordinary differential equations and solved by a tridiagonal matrix algorithm (TDMA) to obtain fluxes and the radiative source term.

While the six-flux model has the advantage of simplicity and being computationally cheap, it does suffer from a number of limitations, including:

- In case of transparent gases, radiation passes from one surface to another without affecting the gas. The six-flux model will not yield very accurate results, since its transmission occurs in coordinate directions only, neglecting the oblique effects.

- No interactions, apart from scattering, arise between the radiation fluxes in the different coordinate directions.

- The model is not readily extended to coordinate systems which are neither Cartesian nor cylindrical-polar.

8.17.2.2 Discrete transfer radiation model (DTM)

The discrete transfer method (DTM) is one of the widely used methods to solve radiative transfer problems with participating medium. It combines advantages of the Monte Carlo method, flux methods and zonal methods (Lockwood and Shah, 1981). One advantage is that this method can be used to calculate the intensity distribution in arbitrary shaped, three dimensional complex geometries. A detailed formulation of the DTM can be found in (Deiveegan, 2011).

The discrete transfer method is based on solving the RTE for some representative rays fired from the boundaries and solid surfaces in the domain. Rays are fired from surface elements into a finite number of solid angles that cover the radiating hemisphere about each element and the main assumption of the DTM is that the intensity through the solid angle is approximated by a single ray. The number of rays and distribution of directions are chosen in advance. The figure below illustrates the ray selection and angular discretisation. In the DTM method the RTE is solved for each ray from one solid boundary to another solid boundary in the geometry. Rays are fired from solid surface boundaries and traced through the volume. The calculation of the radiation source term is based on the distance travelled in each control volume. At the boundaries, radiative heat transfer boundary conditions are used to determine the intensity of rays fired from that surface area. As the correct initial intensities are unknown at the start of the calculation, the procedure is iterative until correct radiative intensities are resolved. If ray intersection data is saved either in memory...
or on disk, then no ray tracing is required after the first iteration; available ray data can be readily used, making the process very efficient.

Figure 8.13: Selection of ray directions for the DTM

The shooting and tracing (Deiveegan, 2011) gives the knowledge about the distance travelled by the ray ($x_{len}$) and the number of segments for each ray ($N_{seg}$). A ray is traced in each direction through the radiation space until it strikes another surface. Then, starting from that surface, the ray is followed back to its origin, while solving for the intensity distribution along its path with the recurrence relation:

$$I_{n+1} = I_n e^{-\beta x_{len}} + S[1 - e^{(-\beta x_{len})}], \quad 1 \leq n \leq (N_{seg} - 1),$$  

(8.150)

where $I_n$ is the intensity of the ray at the entry of the control volume, $I_{n+1}$ is the intensity of the ray at the exit of the control volume, $\beta$ is the extinction coefficient (absorption coefficient + scattering coefficient) and $S$ is the source function. The source function and extinction coefficient are assumed to be constant over the interval $x_{len}$. This procedure is repeated for all rays leaving the given wall cell.

The intensities calculated from the above equation are assumed to be constant over each finite solid angle $\delta \Omega$, such that the incident radiative heat flux is obtained as

$$q_i(r_p) = \sum_{k=1}^{N_{rays}} I_{i,k} \cos \theta_k \sin \theta_k \sin d\theta_k d\phi_k.$$  

(8.151)

This incident flux, together with the temperature and the surface emissivity, are assumed constant over the entire element $S_e$. Then the outgoing intensity is everywhere $I_0 = q_0/\pi$, where the emitted flux has been defined as

$$q_o = (1 - \varepsilon_w)q_i + \varepsilon_w \sigma T^4_\infty.$$  

(8.152)

Hence, for those rays originating from surface elements that strike $S_e$, an initial intensity $I_0$ is used in the recurrence relation shown below. Since $q_o$ depends on the value of $q_i$, an iterative solution is required, unless all the surfaces are black. The net surface heat flux over of the surface elements is then found by restating the equation:

$$q_s = q_o - q_i.$$  

(8.153)

An equation is also required for the divergence of the radiative heat flux ($\nabla \cdot q_R$ in the energy equation) in each volume element. Each ray is considered as a beam of radiative energy, and the heat source associated with its passage through a volume $n$ is

$$Q_{Gn} = (I_{n+1} - I_n)A_n \cos \theta_n \sin \theta_n \sin d\theta d\phi,$$  

(8.154)
where $A_n$ is the area of the surface element from which the ray was emitted and it is assumed that the intensity is constant over the finite solid angle. When the beam of energy associated with a ray only partially intersects a volume then the actual source is a fraction of that in above equation. However, complex source sharing calculations are avoided by simply lumping all the energy $Q_g$ into only those volume elements cut by the central ray path $\hat{s}$; this saves considerable computational effort without significant loss in accuracy. Summing the individual source contributions from all the $N_{\text{rays}}$ rays passing through a volume element, and then dividing this value by its volume, $V_n$, gives the divergence of radiative heat flux as:

$$\nabla \cdot q_R = \sum_{n=1}^{N_{\text{cells}}} \frac{Q_{g_n}}{V_n}. \quad (8.155)$$

The flux divergence is assumed to be constant over each volume element, as for other radiative properties.

Finally, the solution of equation $q_s$ requires a value for the source function $S$. The source function for absorption-only problems is simply $S = I_B$, where $I_B$ is the black body intensity of the medium. For problems involving scattering the source function is approximated as:

$$S = (1 - \omega) I_B[T] + \frac{\omega}{4\pi} \sum_{n=1}^{N_{\text{cells}}} (I_{\text{avg}})_n \Phi_n \delta\Omega_n,$$

where the average intensity is taken as the arithmetic mean of the entering $I_n$ and leaving $I_{n+1}$ radiant intensities for each ray passing through the sub volume within the finite solid angle. This finite solid angle is evaluated for each $d\theta, d\phi$ by angular subdivision as:

$$\delta\Omega_i = 2 \sin \theta_i \sin \left(\frac{\delta\theta}{2}\right) \delta\phi.$$

The source function for an isotropically scattering medium is then found by setting $\Phi = 1$.

### 8.17.2.2.1 Output from the DTM

In general, for temperature-specified problems and source-specified problems, the required results are the heat flux at various walls and temperatures at various points of the medium, respectively. These results are calculated and stored in a DTM Flux.txt file for post processing. The net surface heat flux is calculated by

$$q_s = q_0 - q_i,$$

where the subscripts $o$ and $i$ represent outgoing and incoming respectively. Additionally, the divergence of the heat flux or gas emissive power is obtained as

$$\nabla \cdot q_R = \sum_{n=1}^{N_{\text{cells}}} \frac{Q_{g_n}}{V_n}, \quad (8.159)$$

where the source contribution for a cell is:

$$Q_{g_n} = (I_{n+1} - I_n) A_n \cos \theta_n \sin \theta_n \sin \theta \sin \phi.$$

The results are stored in files for further post processing. Detailed validation of this model can be found in (Deiveegan, 2011).

The primary advantages of the DTM are:

- The DTM is numerically exact and geometrically flexible.
- It can be used to solve conjugate heat transfer problems.
- Very accurate for a wide range of optical thicknesses.

The main limitation of the DTM is:

- Solving a problem with a large number of rays is CPU-intensive.
8.17 Fire model

The computing time and memory requirements of the DTM are significant, especially for cases with large domains. The resource usage can be reduced by employing the DTM only for near-field calculations while using the multi-point source model for far-field calculations. The DTM domain can be selected by using one of following three settings: Automatic (Default), Automatic min, Full. The DTM was parallelized using Ray decomposition parallelization (RDP). In RDP, the total number of rays fired from the boundary of the domain is split into a number of subsets equal to the number of processors, and each subset is assigned to a processor. Also new fast Ray/Axis-aligned bounding boxes (AABBs) intersections using the slab method employed for DTM ray tracing.

8.17.2.3 Far-field radiation model

In this model, radiation is assumed to be emitted from a number of point sources that lie along the length of the flame and emit in all directions. In general, the multi point source model can be expressed as (Mumby, 2010):

\[ q = \sum_{i=0}^{n} \tau_i Q_i \frac{4\pi S_i^2}{r_i}, \]  

(8.161)

where the subscript \( i \) refers to a variable that depends on the distance between the target and the \( i \)th point source on the flame. \( S \) is the absolute distance from the point source to the point target and \( \tau \) is the atmospheric transmissivity along that path.

Figure 8.14: Multi-point source model.

The estimation of the strength of the point sources and its location plays a critical role. The first step in determining the strength of the point sources is to find the longest axis of the flame. Along the longest axis, each grid plane perpendicular to the axis is considered to have point sources. The strength of the point sources \( (Q_i [W]) \) are calculated by adding radiative source terms in the warmest cell of each plane. For multiple point sources along the flame’s trajectory, each point radiates a proportion of the total emitted power of the flame, and each contributes to the incident radiation on the target. The approach is to arrange the points on the flame into \( n \) points each radiating \( \frac{1}{n} \) of the total emissive power. Moorhouse and Pritchard (1982) presented the following relationship to approximate the transmissivity of infrared radiation from hydrocarbon flames through the atmosphere (valid for distances up to 300 m):

\[ \tau = 0.998^S. \]  

(8.162)
The far-field calculations are performed for cell centers, surfaces and monitor points. For surfaces and monitors, the shadow effect of the geometry is considered. The variable QRAD represents the radiative flux with shadow (geometrical blockage) effects, while the variable QRADFF ignores the presence of geometry. Unless results are required in cells without surfaces (which is only possible with QRADFF), it is recommended to only use the QRAD output variable as it accounts for shadow effects.

8.17.2.4 Radiative properties model

The absorption coefficient is calculated from the temperature, pressure, the mole fractions of \(CO_2\) and \(H_2O\) and the path length in the participating medium. In FLACS-Fire, \(CO_2\) and \(H_2O\) have been considered as the participating gasses that absorb and emit radiation depending on the local mixture temperatures. There are two types of absorption coefficient models available with FLACS-Fire: uncoupled and coupled. The following models are implemented in FLACS-Fire to evaluate gaseous radiative properties:

- Constant: User-specified constant absorption coefficient.
- Danish-Coupled-WSGGM: Coupled/line of sight WSGGM based on Yin et al., (2013).

The coupled/line of sight models use the true path length of a ray crossing a computational cell to calculate an absorption coefficient. In contrast, the uncoupled version of the DTM, which pre-calculates an absorption coefficient, does not use the true path length. In these models the mean beam length is used as path length. The mean beam length is a constant value proportional to the ratio of volume to surface area.

- Domain based: Based on whole domain.
- Cell based: Based on local cell size.
- Max ray length: Based on maximum length of the domain.
- LSLIM: Based on characteristic geometrical dimension (LSLIM).

8.17.3 Soot model

A well-predicted soot level is important for heat transfer by radiation. Soot is a difficult phenomenon within combustion science, due to the lack of knowledge about the soot formation and growth. Both simplified and detailed models are found in the literature (Kennedy, 1997). However, independent of their complexity and degree of modelling and empiricism all models need some form of adjustments to give good predictions (Haynes & Wagner, 1981).

Currently there are two approaches for the modelling of soot in FLACS-CFD, the Conversion Factor Model (CFM) and the Formation-Oxidation model (FOX). Details about these model approaches follow.

8.17.3.1 Fixed conversion soot model

In the fixed Conversion Factor Model (CFM) a certain fraction of the fuel carbon is converted to soot directly. This conversion depends on the fuel composition and independent of equivalence ratio, temperature, time, etc. The only input to the model is the soot yield, which gives the fraction of carbon from the fuel that is converted to soot. The soot yield for different fuels are listed in the table Soot yield for fuels in FLACS-CFD, where most of the values are from Kent (1986).
Table 8.5: Soot yield for fuels in FLACS-CFD

<table>
<thead>
<tr>
<th>Species</th>
<th>Soot yield</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methane</td>
<td>0.7%</td>
</tr>
<tr>
<td>Ethane</td>
<td>2.0%</td>
</tr>
<tr>
<td>Propane</td>
<td>9.0%</td>
</tr>
<tr>
<td>Butane</td>
<td>10.0%</td>
</tr>
<tr>
<td>Pentane</td>
<td>10.0%</td>
</tr>
<tr>
<td>Hexane</td>
<td>10.0%</td>
</tr>
<tr>
<td>Heptane</td>
<td>12.0%</td>
</tr>
<tr>
<td>Octane</td>
<td>12.0%</td>
</tr>
<tr>
<td>Nonane</td>
<td>12.0%</td>
</tr>
<tr>
<td>Decane</td>
<td>12.0%</td>
</tr>
<tr>
<td>Hendecane</td>
<td>12.0%</td>
</tr>
<tr>
<td>Dodecane</td>
<td>12.0%</td>
</tr>
<tr>
<td>Ethylene</td>
<td>12.0%</td>
</tr>
<tr>
<td>Propylene</td>
<td>16.0%</td>
</tr>
<tr>
<td>Acetylene</td>
<td>23.0%</td>
</tr>
</tbody>
</table>

8.17.3.2 Oxidation Formation Soot Model (FOX)

In the Formation-Oxidation (FOX) model for soot there are two source terms in the transport equation for soot. One term models the formation of soot and the other models the oxidation (combustion) of soot. The formation of soot is a slow process, in particular the formation of nuclei requires a rich mixture, a certain temperature, and time. When the nuclei are present soot can be formed rather quickly. For the modelling of nuclei formation and oxidation, one more transport equation is required. In the current modelling of soot with the FOX model, nuclei formation and oxidation has not been included.

The transport equation for the soot mass fraction is written as:

\[
\frac{\partial \tilde{Y}_{\text{soot}}}{\partial t} + \frac{\partial \tilde{Y}_{\text{soot}} \tilde{u}_j}{\partial x_j} = \frac{\partial}{\partial x_j} \left( \rho D \frac{\partial \tilde{Y}_{\text{soot}}}{\partial x_j} \right) - \frac{\partial}{\partial x_j} \left( \rho \tilde{Y}_{\text{soot}}' \tilde{u}'_j \right) + \tilde{\omega}_{\text{soot}}. \tag{8.163}
\]

The source term in the transport equation for soot is the sum of a formation and an oxidation term:

\[
\tilde{\omega}_s = \tilde{\omega}_{s,f} + \tilde{\omega}_{s,ox}. \tag{8.164}
\]

The model for formation of soot is modelled as (Khan & Greeves, 1974):

\[
\tilde{\omega}_{s,f} = \begin{cases} 
C_{l} p X_{\text{fuel}} (ER)^3 \exp \left( E_l / RT \right) & \text{for } ER \in (1.67, 3) \\
0 & \text{else,} \end{cases} \tag{8.165}
\]

where \( C_{l} = 1.5 \) and \( E_l / R = 20000K \) for most hydrocarbons. Magnussen & Hjertager (1976) modelled the oxidation of soot similar to oxidation of fuel, cf. Eddy dissipation concept. Hence, the oxidation of soot is modelled as:

\[
\tilde{\omega}_{s,ox} = - \frac{\dot{m}_Y}{1 - \gamma^* \chi} \tilde{Y}_{\text{min}} \tag{8.166}
\]

\[
\tilde{Y}_{\text{min}} = \min \left( \tilde{Y}_s, \frac{1}{r_s} \tilde{Y}_{\text{ox}} \right), \tag{8.167}
\]

where \( r_s = 8/3 \) is the stoichiometric amount of oxidant for soot combustion. Currently, the upper limit for the mass fraction of soot, when using the FOX model, is given by the soot yield value, cf. Table Soot yield for fuels in FLACS-CFD.
8.17.4 Conduction

8.17.4.1 Heat equation for walls

The heating of a wall element with uniform properties is given by

\[ mc \frac{\partial T}{\partial t} = \dot{Q}_w, \]  
(8.168)

where \( m \) is the mass and \( c \) is the specific heat of the element and \( \dot{Q}_w \) is the sum of the heat transported over an element boundary.

Assuming that the element is a box and that heat is transported by conduction to the neighbouring elements and by convection and radiation to the surrounding flow, this gives:

\[ \dot{Q}_w = \sum_{i=1}^{6} \dot{Q}_{w,i}, \]  
(8.169)

where

\[ \dot{Q}_{w,i} = \dot{Q}_{\text{cond},i} + \dot{Q}_{\text{conv},i} + \dot{Q}_{\text{rad},i}. \]  
(8.170)

The area \( A_{c,i} \) is in contact with the neighbouring wall element and the area \( A_{g,i} = A_i - A_{c,i} \) is in contact with the flow. Hence, the conductive heat transfer is given by Mills (1995):

\[ \dot{Q}_{\text{cond}} = A_{c} k_c \frac{\partial T}{\partial x}, \]  
(8.171)

where the effective conductivity between material 1 and 2 is given by:

\[ k_c = \frac{2k_{c,1}k_{c,2}}{k_{c,1} + k_{c,2}}. \]  
(8.172)

The convective heat transfer is given by:

\[ \dot{Q}_{\text{conv}} = A_{g} \dot{q}_{\text{conv}}, \]  
(8.173)

where \( \dot{q}_{\text{conv}} \) is calculated by using the law of walls for turbulent flows. In the law of walls, a dimensionless wall distance \( y^+ \) is defined by:

\[ y^+ = \frac{C\mu k^{1/2}y}{\nu}. \]  
(8.174)

Then the convective heat transfer is written as:

\[ \dot{q}_{\text{conv}} = \rho g h^{1/4} k^{1/2} e_{p,g} \left( T_i - T_w \right), \]  
(8.175)

where

\[ T^+ = \begin{cases} \Pr y^+ & \text{if } y^+ < E^+ \\ E^+ \Pr + \frac{\Pr}{\kappa} \ln \frac{y^+}{E^+} & \text{if } y^+ \geq E^+ \end{cases} \]  
(8.176)

with the laminar Prandtl number for air, \( Pr = 0.7 \) (Kays & Crawford, 1993, Appendix A.1) and the turbulent Prandtl number is \( Pr_t = 0.7 \). The specific heat of the gas, \( e_{p,g} \), is evaluated at the mean temperature \( T_m = \frac{1}{2}(T_i + T_w) \). The constants in above equations are \( E^+ = 9.0, \kappa = 0.42, \) and \( C\mu = 0.09 \). The walls are assumed to absorb nearly all of the radiation:

\[ \dot{Q}_{\text{rad},w} = A_g e_w \left( \dot{q}_{\text{rad}} - \sigma T^4 \right), \]  
(8.177)

where \( e_w = 0.9 \) in FLACS-CFD.

The heat equation is solved explicitly in time and instabilities may appear when very high heat loads are applied.
8.18 Porosity calculations and geometry counting

8.17.4.2 Wall elements

You can specify boxes with uniform thermal properties. If a box crosses a FLACS-CFD fluid cell, it is divided at the fluid cell boundary such that a reasonable resolution for the conduction is achieved. You may specify boxes with dimensions smaller than the grid. This might be useful when building composite materials or walls with insulation. An example is shown in the figure below (right). There are two predefined wall materials, steel and concrete, for which thermal properties are given in Table material properties. In addition, ten materials can be specified, see section Thermal Active Walls. A simplification is made by assuming constant material properties. It should be mentioned that specific heat and conductivity may vary considerably with temperature (Mills, 1995).

![Figure 8.15: Walls and fluid grid. Left: A wall element divided by the fluid cells. Right: Three wall elements in each fluid cell.](image)

**Table 8.6: Predefined material properties and some examples**

<table>
<thead>
<tr>
<th>Material</th>
<th>Density [kg/m³]</th>
<th>Specific heat [J/kg K]</th>
<th>Conductivity [W/(m K)]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Predefined</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steel</td>
<td>7840</td>
<td>450</td>
<td>50</td>
</tr>
<tr>
<td>Concrete</td>
<td>2100</td>
<td>880</td>
<td>1.4</td>
</tr>
<tr>
<td>Examples</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aluminium</td>
<td>2702</td>
<td>903</td>
<td>237</td>
</tr>
<tr>
<td>Bricks</td>
<td>1600</td>
<td>840</td>
<td>0.69</td>
</tr>
<tr>
<td>Polystyrene</td>
<td>30–60</td>
<td>1210</td>
<td>0.028</td>
</tr>
<tr>
<td>Polyurethane</td>
<td>70</td>
<td>1120</td>
<td>0.026</td>
</tr>
<tr>
<td>Stainless steel</td>
<td>8000</td>
<td>480</td>
<td>15</td>
</tr>
</tbody>
</table>

8.18 Porosity calculations and geometry counting

This section describes the utilities FGC (Flacs Geometry Calculator), Porcalc, cofile (deprecated) and cofile2. FGC and Porcalc are used to calculate the area and volume porosities (blocking of each grid cell by the geometry elements in this cell). cofile2 is used to count the number of primitives (e.g. boxes and cylinders) in a geometry, and compute length, area and volume of the primitives. This report can be used for geometry verification and in connection with the anticipated congestion method.
8.18.1 FGC (Flacs Geometry Calculator)

FGC is used to calculate area and volume porosities. The porosity is a value between 0 and 1 for the mean blockage of a control volume. A value of 0 means that the control volume is completely blocked, a value of 1 means that the control volume is completely open. The volume porosity is the mean blockage of the inside volume of the control volume and the area porosity the blockage of the surface area of the control volume. There are 3 area porosities for each control volume, one for each of the surfaces on the negative axis sides of the control volume box.

8.18.1.1 Gaps in area porosity

In some cases, such as when working with flat rotated boxes, the area porosities might have unexpected gaps (i.e. be less than 1). An example of this is shown in the figure below. This figure shows the area and volume porosities for the same geometry where FGC have been run with and without the closeGaps feature enabled. Trouble areas are highlighted with green circles.

Note:

It is important to note that this feature have seen limited validation, and should only be used when the area porosities are clearly wrong. The resulting porosities should always be verified by you.

8.18.1.2 Trace area porosity

The trace area porosity function is used by FGC to ensure "water tight" area porosities, and avoid gaps. It works by tracing the outline of one or more objects, and blocking grid cell faces that lie along the outline. It is by default used on large (>4 grid cells), zero thickness objects, but can be overridden in the primitive properties.

8.18.1.3 Terrain in FGC

Terrains that are stored within the CO file or in terrain files (.tri) are supported by FGC and treated in the same way that Porcalc treats terrain.
8.18 Porosity calculations and geometry counting

8.18.2 cofile

The FLACS-CFD utility cofile is used to report out the amount of boxes and cylinders in the geometry in a cofile.

All geometry elements are counted by default, but filters can be set such that only elements inside a given region or with a certain colour are considered. Cylinders and boxes are counted and reported separately. The report shows the length and projected areas of the elements within certain predefined or user-defined classes.

8.18.2.1 Counting algorithm in cofile

The following gives an overview of the algorithm that counts boxes and cylinders in cofile.

8.18.2.1.1 Algorithm to count cylinders

1. Filter cylinders to match given colour hues (option hues=...)
   - cylinders with other colour hues than specified are discarded

2. Filter cylinders to match given diameter range (option mindia=... and maxdia=...)
   - cylinders with diameter smaller than mindia or bigger than maxdia are discarded
   - cylinders with zero length are automatically discarded (this could be undesired and may be changed in future versions)

3. Clip cylinders to fit within given region (option region=...)
   - cylinders that fall outside the region are discarded
   - cylinders that fall inside the region (partially or fully) will be counted
   - the length of cylinders extending outside the region is cut away

4. Split the cylinders into 3 separate sets, one for each axis direction X, Y and Z, and calculate lengths and projected surface areas, remove overlapping contributions
   - each of the 3 sets are treated independently
   - the length of overlapping cylinders are chopped into smaller pieces, only visible pieces remain
   - the length of cylinder pieces that are partly visible in the radial direction is scaled according to the visible diameter fraction (\(\text{Length}_{\text{scaled}} = \text{Length} + \frac{D_{\text{visible}}}{D}\))
   - overlaps between cylinders and other object types are ignored

5. Sort length and area contributions into diameter classes
   - separate counts for cylinders in X, Y and Z direction

8.18.2.1.2 Algorithm to count boxes

1. Filter boxes to match given colour hues (option hues=...)
   - boxes with other colour hues than specified will be discarded

2. Clip boxes to fit within given region (option region=...)
   - boxes that fall outside the region are discarded
   - boxes that fall inside the region (partially or fully) will be counted
   - the part of boxes extending outside the region is cut away; this changes the box shape (\(\text{Length} > \text{Height} > \text{Width}\))

3. Calculate lengths and surface areas, remove overlapping contributions
• overlaps between boxes and other object types are ignored

4. Sort length and area contributions into diameter classes

• the 'length' of a box is defined as the size of the longest side
  – Length > Height > Width
  – Length/2 falls into diameter class where Diameter = Height
  – Length/2 falls into diameter class where Diameter = Width

• the length of partially visible boxes is scaled according to its 'remaining volume' \( \text{Length}_{\text{scaled}} = \frac{\text{Length} \times V_{\text{remain}}}{V} \)

• the remaining volume of a box is calculated as its original volume minus intersecting volumes cut away by boxes appearing later in the list of boxes

• the surface areas are sorted into diameter classes in the following way
  – \( A_x \) uses Diameter = \min(D_y, D_z)
  – \( A_y \) uses Diameter = \min(D_x, D_z)
  – \( A_z \) uses Diameter = \min(D_x, D_y)

8.18.2.1.3 Counting of objects positioned side-by-side

Objects that are placed side-by-side are counted separately in terms of length. For two or more boxes placed adjacent to each other along the length direction, this may lead to a larger total box length compared to a single box occupying the same volume.

Figure 8.17: Representing a box by two adjacent boxes of half the thickness will lead to the double length (here along the z-axis) being reported by cofile.

8.18.2.1.4 Overlapping cylinders and boxes in the counting algorithms: In the algorithms given above, some overlaps are said to be ignored. The current paragraph addresses which overlaps are disregarded to prevent or reduce double counting.

For **aligned objects of the same type**, the overlap is calculated and disregarded for the statistics (the overlap calculation is an approximation to keep the calculation times reasonable).

**Example:** Two aligned 4" pipes of length 100 m overlapping by 25 % will contribute to the cylinder length as \( 2 \times 75 \text{ m} = 150 \text{ m} \).

For **non-aligned objects of the same type**, the overlap is normally relatively small and therefore it is not disregarded.

**Example:** Two perpendicular 20" pipes of length 1 m are counted with the length \( 2 \times 1 \text{ m} = 2 \text{ m} \).

For **objects of different types** (cylinder vs. box), even when they are aligned, the overlap is harder to quantify and attribute; therefore the contributions to cylinders and boxes are fully counted. See also the following figure and table for some examples.
8.18 Porosity calculations and geometry counting

Figure 8.18: Different types of overlap between objects.

Table 8.7: cofile results for (partly) overlapping objects.

<table>
<thead>
<tr>
<th></th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length cylinders [m]</td>
<td>15</td>
<td>15</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>Length boxes [m]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Cylinder Ac [m²]</td>
<td>32</td>
<td>32.4</td>
<td>43.1</td>
<td>21.6</td>
</tr>
<tr>
<td>Box Ac [m²]</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>42</td>
</tr>
</tbody>
</table>

8.18.2.1.5 Known issues and problems

- Calculations with very large domains (10km or more) may be slightly inaccurate
- The sum of areas of cylinders may be inaccurate in complex overlap situations
- Overlaps between cylinders, boxes and ellipsoids are ignored
- Object porosities are ignored

8.18.3 cofile2

The new cofile2 utility also reports the number of basic primitives in the geometry, but has a number of improvements compared to the old cofile utility, e.g. it also reports the volume in addition to lengths and area.

The reported surface area of cylinders is the true surface area, and not the projected area in x-, y-, and z-direction. The computation of surface area in cofile2 takes into account overlapping primitives, which means that the reported surface area is more correct than what was reported in the older cofile utility. This applies to both union and left-difference operations between primitives and groups of primitives.

In order to compute the surface area of the geometry, cofile2 converts all the solids into surface representations, and uses a binary space partitioning (BSP) algorithm to split overlapping surfaces and discard parts that are inside other primitives.

Compared to the old cofile utility the cofile2 utility computes the length of overlapping primitives differently, and will not scale the length according to the degree of overlap, e.g. two overlapping boxes of length 10m will report a total length of 20m.

8.18.3.1 Known issues and problems with cofile2

- The accuracy of computing surface area and volumes leads to a longer computational time than the old cofile
• The sum of areas of cylinders may be inaccurate in complex overlap situations
• Object porosities are ignored

8.19 Jet flame shape

To configure the simulation grid appropriately for jet fire simulations, it is necessary to estimate the likely maximum dimensions of the flame and the flame lift-off. These can be estimated using the model from Chamberlain, 1987 to calculate the dimensions of a frustrum that encloses the flame. This method is used for the automatic grid refinement in CASD. Note that this method does not account for impingement and so it may be more appropriate to estimate the flame shape from an initial coarse-resolution simulation for congested scenarios.

The steps to calculate the frustrum length, $FL$, frustrum width $FW$, and flame lift-off, $b$, are outlined briefly below. For the purposes of configuring the grid, $FL$ and $FW$ can be interpreted as the estimated flame length and width, respectively. It is recommended that the frustrum length is extended by 10% and the width is doubled (to give a conservative estimate of the flame dimensions) before these estimates, and the estimated lift-off, $b$, are used to configure the Recommended grid configuration. It should be noted that these calculations provide estimates of the flame length along the flame center line, and the flame width perpendicular to this. If the flame center line is not parallel to a grid axis, then the estimated dimensions should be projected onto the grid before being used to determine the appropriate core domain extent (this projection is handled when the calculations are implemented in the automatic grid refinement in CASD).

$D$ is the diameter of the expanded leak area in m
$u_{jet}$ is the exit velocity of the jet from the leak (this can be calculated using the Leak wizard in CASD) in m/s
$u_{wind}$ is wind speed in m/s
$W$ is the mass fraction of the leaked fuel in a stochiometric mix with air
$\theta$ is the angle between the hole axis and the horizontal in degrees

1. Solve iteratively for $Y$:

$$0.024 \times \left( \frac{9.81 \times D}{u_{jet}^2} \right)^{\frac{1}{3}} Y^{\frac{5}{2}} + 0.2 Y^{\frac{7}{2}} - \left( \frac{2.85}{W^2} \right)^{\frac{1}{3}} = 0 \quad (8.178)$$

2. Calculate the flame length in still air: $L_0 = YD$

3. Calculate the length of the flame, $L$:

$$L = L_0 \times \left( 0.5e^{(-0.4u_{wind})} + 0.49 \times (1.0 - (6.07 \times 10^{-3}) \times (\theta - 90)) \right) \quad (8.179)$$

4. Calculate the Richardson number, $R_i$:

$$R_i = \left( \frac{9.81}{D^2 \times u_{jet}^2} \right)^{\frac{1}{3}} \times L_0 \quad (8.180)$$

5. Calculate the tilt angle, $\alpha$:

- If $\frac{u_{wind}}{u_{jet}} < 0.05$:

$$\alpha = (\theta - 90) \times \left( 1 - e^{(-25.6 \times \frac{u_{wind}}{u_{jet}})} \right) + \frac{8000 \times \frac{u_{wind}}{u_{jet}}}{R_i} \quad (8.181)$$
8.20 Maximum pool area and flame shape for pool scenarios

This section describes how to compute the maximum pool area and flame shape for pool scenarios, as required for the Recommended grid configuration. These calculations are implemented in the Automatic grid refinement for pool fires in CASD. Note that this method does not account for impingment and so it may be more appropriate to estimate the flame shape from an initial coarse-resolution simulation for congested scenarios.

8.20.1 Inputs required for flame shape calculation

- If $\frac{u_{\text{wind}}}{u_{\text{jet}}} > 0.05$:

  $$\alpha = (\theta - 90) \times \left(1 - e^{\left(-25.6 \times \frac{u_{\text{wind}}}{u_{\text{jet}}}\right)}\right) + \left(\frac{134 + 1726 \times \left(\frac{u_{\text{wind}}}{u_{\text{jet}}} - 0.026\right)^{\frac{1}{2}}}{R_i}\right)$$  \hspace{1cm} (8.182)

6. Calculate the flame lift-off, $b$:

- If tilt angle ($\alpha$) $< 0.01$:
  $$b = 0.2 \times L$$  \hspace{1cm} (8.183)

- If tilt angle ($\alpha$) $\approx 180$:
  $$b = 0.015 \times L$$  \hspace{1cm} (8.184)

- Otherwise:
  $$K = 0.185 \times e^{\left(20 \times \frac{u_{\text{wind}}}{u_{\text{jet}}}\right)} + 0.015$$  \hspace{1cm} (8.185)
  $$b = L \times \frac{\sin(K \times \alpha)}{\sin \alpha}$$  \hspace{1cm} (8.186)

7. Calculate the flame lift-off, $b$:

- If tilt angle ($\alpha$) $< 0.01$:
  $$b = 0.2 \times L$$  \hspace{1cm} (8.187)

- If tilt angle ($\alpha$) $\approx 180$:
  $$b = 0.015 \times L$$  \hspace{1cm} (8.188)

- Otherwise:
  $$K = 0.185 \times e^{\left(20 \times \frac{u_{\text{wind}}}{u_{\text{jet}}}\right)} + 0.015$$  \hspace{1cm} (8.189)
  $$b = L \times \frac{\sin(K \times \alpha)}{\sin \alpha}$$  \hspace{1cm} (8.190)

8. Calculate the frustrum length, $FL$:

  $$FL = (L^2 - b^2 \times \sin^2 \theta)^{\frac{1}{2}} - b \times \cos \alpha$$  \hspace{1cm} (8.191)

9. Calculate the frustrum tip width, $FW$:

  $$FW = L \times \left(0.18e^{-1.5 \times \frac{u_{\text{wind}}}{u_{\text{jet}}}} + 0.31\right) \times \left(1.0 - 0.47e^{-25 \times \frac{u_{\text{wind}}}{u_{\text{jet}}}}\right)$$  \hspace{1cm} (8.192)
Table 8.8: cofile results for (partly) overlapping objects.

<table>
<thead>
<tr>
<th>Burning Rate of Fuel</th>
<th>Kinematic Viscosity of Air</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density of Air</td>
<td>Wind speed and direction</td>
</tr>
</tbody>
</table>

Non spreading pool
- Pool diameter
- Pool bund height
- Pool position (on/above/below ground, see figure)

Spreading pool
- Tank dimensions, volume
- Liquid density of fuel
- Hole diameter
- Pool thickness
- Pool Position (On/above/below ground, see figure below)

Figure 8.19: Position of pool relative to substrate.

8.20.2 Estimating the maximum pool diameter for unconfined pools

For confined pools the pool diameter is equal to the bund diameter. For unconfined pools, depending on flow conditions, the pool will grow until it reaches its maximum size. The flame shape for confined pools is estimated by assuming that this maximum size has been reached. Inputs required for estimating the maximum pool diameter:

- Tank volume, diameter, area (\(V_{tank}, D_{tank}, A_{tank}\))
- Liquid density of fuel (\(\rho_{liqfuel}\))
- Discharge coefficient (Co)
- Hole Diameter or orifice diameter (\(D_0\))
- Pool Thickness (\(P_{thickness}\)), this value should be assumed if not available
- Burning rate (\(m''\))
- \(k \times \beta\), for the fuel
- \(g=9.81 \text{ m/s}^2\)

Fuel mass (\(m_{fuel}\))

\[
m_{fuel} = V_{tank} \times \rho_{liqfuel}(kg)
\]  

Initial height of fuel above release point (\(H_i\))

\[
H_i = \frac{V_{tank}}{A_{tank}}(m)
\]  

Fuel mass (\(m_{fuel}\))
8.20 Maximum pool area and flame shape for pool scenarios

Hole Surface Area \((A_0)\)

\[ A_0 = \frac{\pi}{4} \times (D_0)^2 \text{(m}^2\text{)} \tag{8.195} \]

Initial outflow velocity \((V_0)\)

\[ V_0 = C_0 \times (2 \times g \times H_i)^{1/2} \text{(m/s)} \tag{8.196} \]

Acceleration constant \((a)\)

\[ a = (C_0 \times A_0 / A_{tank})^2 \times g \text{(m/s}^2\text{)} \tag{8.197} \]

Dimensionless parameter \((\beta')\)

\[ \beta' = \frac{(a \times P_{thickness} \times \rho_{liqfuel})}{(V_0 \times m''')} \tag{8.198} \]

Time at which pool diameter will be maximum \((t_{max})\) seconds

\[ \tau_{max} = \ln \left( \frac{1 + \beta'}{\beta'} \right) \tag{8.199} \]

\[ t_{max} = \frac{\tau_{max} \times \rho_{liqfuel} \times P_{thickness}}{m''} \tag{8.200} \]

Maximum Pool Diameter \((D_{pool, m})\)

\[ \theta_{max} = 1 + (\beta' \times [(1 - \tau_{max}) - (1 + \beta')] \times e^{-\tau_{max}} \tag{8.201} \]

\[ D_{max} = 2 \times \left[ \frac{\theta_{max} \times \rho_{liqfuel} \times A_0 \times V_0}{(m'''' \times \pi)} \right]^{1/2} \tag{8.202} \]

8.20.3 Flame shape

The flame shape can be determined by the following input parameters:

- Pool area
- Burning rate
- Wind velocity

Flame structure comprises:

- Flame length
- Flame tilt
- Flame drag

Pool Diameter and Pool Area

- For Circular Pool, “Pool Area” is based on the pool diameter
- For square or rectangular pool, if the length is less than twice the width, then an equivalent diameter must be calculated.
• If volume and thickness of the pool is known, then

\[ D = \left( 4 \times \left( \frac{V}{\pi \times \delta} \right) \right)^{1/2} \]  

(8.203)

\( V = \) Volume of the released liquid in m\(^3\)
\( \delta = \) Thickness of the pool, in m

**Characteristic Wind Velocity** \((U_c)\)

\[ U_c = g \times m^* \times D/\rho_{air} \]  

(8.204)

\( U_c = \) characteristic wind velocity, in m/s
\( g = \) Gravitational acceleration 9.81 m/s\(^2\)
\( m^* = \) Burning flux in still weather conditions, in kg/(m\(^2\)s)
\( D = \) Pool diameter, in m

**Scaled Wind Velocity** \((U^*)\)

\[ U^* = U_w/U_c \]  

(8.205)

\( U_w = \) Wind velocity at height of 10 meters, in m/s

**Average Flame Length** \((L)\)

\[ \frac{L}{D} = 55 \times \left( \frac{m^*}{\rho_{air} \times (g \times D)^{1/2}} \right)^{0.67} \times (U^*)^{-0.21} \]  

(8.206)

\( L = \) Average flame length, in m

**Flame Tilt** \((\theta)\)

\[ Fr_{10} = U_w/(g \times D) \]  

(8.207)

\( Fr_{10} = \) Froude number for wind velocity height of 10 meters

\[ Re = \frac{(U_w \times D)}{v} \]  

(8.208)

\( v = \) Kinematic viscosity of air, in m2/s

\[ \frac{\tan \theta}{\cos \theta} = 0.666 \times (Fr)^{0.333} \times (Re)^{0.117} \]  

(8.209)

\( \theta = \) Tilt angle of the flame, in degrees

In general, if \( \tan \theta/\cos \theta = C \), then \( \theta \) can be calculated by:

\[ \theta = \sin^{-1} \left( \frac{4 \times C^2 + 1)^{1/2}}{2 \times C} \right) \]  

(8.210)

**Flame Drag** \((D')\)

For conical flame

\[ \frac{D'}{D} = 1.6 \times (Fr_{10})^{0.061} \]  

(8.211)

For cylindrical flame

\[ \frac{D'}{D} = 1.5 \times (Fr_{10})^{0.069} \]  

(8.212)
D = Pool Diameter, in m
D’ = Actual elongated flame base diameter, in m

**Flame Triangle**

![Flame Triangle Diagram](image)

Figure 8.20: Triangle, used to estimate extent of secondary core domain.

Under windy conditions, the flame will tend to tilt in the direction of the wind. The flame length and tilt are determined as described above. The flame length is calculated from the pool centre. The flame height is used to set the vertical extent of the core domain appropriately, and the flame base (the projected flame length) is used to set the secondary core domain extent appropriately. If wind velocity = 0 m/s, then the flame will be vertical, and no secondary core domain will be required. Depending on wind velocity, the extent of the secondary core domain in direction of the wind can be computed, which here is represented as $L2_{wind}$. Flame tilt is always measured as the angle between the flame and the vertical axis.

![Wind Direction Diagrams](image)

Figure 8.21: Core and total domain extents in the horizontal plane.

\[
\text{FlameHeight} = \sin(90 - \theta) \times \text{FlameLength} \tag{8.213}
\]
\[
\text{FlameBase} = \cos(90 - \theta) \times \text{FlameLength} \tag{8.214}
\]

Length of the secondary domain from the pool centre, in the direction of the wind.

\[
L2_{wind} = \text{FlameBase} - \text{PoolRadius} \tag{8.215}
\]
Dimensions of the secondary core domain when the wind direction is not along the X-axis

\[ L_{2windX} = \sin \theta_{wind} \times L_{2wind} \]  
\[ L_{2windY} = \cos \theta_{wind} \times L_{2wind} \]  

### 8.21 The initial high pressure region for a blast

The gridding recommendations for a blast scenario refer to the initial high pressure region, which is the sphere used to represent the explosive that generates the blast wave in FLACS-Blast.

To calculate the region occupied by the sphere, we use a typical gas production rate, \( V_{ex} \) (\( V_{ex} = 1.35 m^3 kg^{-1} \) or \( 1.62 m^3 kg^{-1} \) for TNT-like or RDX-like detonations, respectively) to calculate the volume of gas produced from the specified explosive mass, \( M_{ex} \):

\[ V_{init} = V_{ex} \times M_{ex} \]  

(8.218)

The temperature, \( T_1 \), and pressure, \( P_1 \), are set to the ambient air properties and then heated to temperature \( T_{ex} \), following the ideal gas law. \( T_{ex} \) is based on the heat of the reaction and the temperature of the combustion products (\( T_{ex} = 3073.15 \) K or \( 4073.15 \) K for TNT-like or RDX-like detonations, respectively). The new pressure, \( P_2 \), is then:

\[ P_2 = \frac{P_1 \times T_{ex}}{T_1} \]  

(8.219)

After heating, the gaseous sphere is adiabatically compressed to a final pressure, \( P_{ex} \) (\( P_{ex} = 808 \) barg or \( 936 \) barg for TNT-like or RDX-like detonations, respectively), and the volume \( V \), is calculated using the gas expansion coefficient, \( \kappa \), which is 1.4 for diatomic gases:

\[ V = V_{init} \times \left( \frac{P_{ex} + 1}{P_1} \right)^{\frac{\kappa - 1}{\kappa}} \]  

(8.220)

A correction is applied for the number of sides on which there is confinement, so that the radius of the sphere, \( R_{hiP} \), is:

\[ R_{hiP} = \left( \frac{n \times 3 \times V}{4 \times \pi} \right)^\frac{1}{3} \]  

(8.221)

where \( n = 2^N \) and \( N \) is the number of dimensions in which the blast is confined on one side only (\( N \) is between 0 and 3). The initial high pressure region is centred at the detonation location and extends by a distance \( R_{hiP} \) from this in every direction.

### 8.22 The high pressure region for a blast wave

The gridding recommendations for a blast scenario refer to the region of the simulation domain where the peak overpressure from the blast wave exceeds 1 barg. This can be estimated from an initial coarse-resolution simulation, or by using the method below, which is based on Baker, 1977. Note that this method does not account for reflections from surfaces (other than the ground), which may increase the blast volume, and so
is only appropriate for scenarios with little or no geometry.

To simulate the blast wave, FLACS-Blast treats the explosive like a bursting balloon, starting with a gaseous sphere at high temperature and pressure with zero velocity. The calculations for the initial high pressure region provide the volume, $V_g$, and pressure, $P_1$, for the sphere. These are used to calculate the available energy, $E_{av}$, setting $\gamma = 1.4$ (following Baker, 1977):

$$E_{av} = \frac{P_1 \times V_g}{\gamma - 1}$$  \hspace{1cm} (8.222)

Following Baker, 1977, the effective blast wave energy, $E_{ex}$, is assumed be double the available energy for a detonation very close to the ground (less than 15 degrees above the horizon, when viewed from the target), so $E_{ex} = 2 \times E_{av}$. Otherwise, $E_{ex} = E_{av}$. To calculate the overpressures, we need the dimensionless distance, $\bar{R}$, which is a function of the ambient atmospheric pressure, $P_a$ and the distance from the blast for which the max overpressure will be calculated, $r_t$:

$$\bar{R} = r_t \times \left(\frac{P_a}{E_{ex}}\right)^{\frac{1}{3}}$$  \hspace{1cm} (8.223)

The non-dimensional side-on pressure of the blast wave, $\bar{P_s}$, is read from Figures 2-18 and 2-19 in Baker, 1977 for the calculated $\bar{R}$. Note that more complex calculations are required to estimate $\bar{P_s}$ for very close to the blast centre and the calculations here will result in a highly conservative pressure estimate for distances close to the blast centre. Note that the motivation for estimating the high pressure region is to determine the core domain extent, which should also include all region(s) of interest, see Recommended grid configuration, and FLACS-Blast is not recommended for scenarios where near-field results are of interest, see limitations in Blast wave propagation.

If detonation occurs above the ground, then $\bar{P_s}$ is adjusted to account for reflections. If $\bar{R} < 1$, then $\bar{P_s} = 2 \times P_s$, otherwise $\bar{P_s} = 1.1 \times P_s$. The peak side-on overpressure, $P_s$, is then given by:

$$P_s = \bar{P_s} \times P_a$$  \hspace{1cm} (8.224)

To estimate the high pressure region, where the blast wave peak exceeds 1 barg, these calculations should be repeated for increasing values of $r_t$ until $P_s$ exceeds 1 barg.

### 8.23 Linux quick reference

This section summarises some relevant information for users that run FLACS-CFD under the Linux operating system. Further information concerning Linux may be found at e.g. [www.linux.org](http://www.linux.org)

#### 8.23.1 Distributions

FLACS-CFD works on most recent Linux distributions. An updated list of distributions, on which FLACS-CFD has been tested, is given in Hardware and Software requirements.

#### 8.23.2 Desktop environments

FLACS-CFD works independently of the Desktop environment. The most popular environments are:

- KDE
- Gnome
8.23.3 Shell
A command shell is command line interface computer program to an operating system (OS). The most popular shells are:

- **bash**, setup file in home directory: .bashrc
- **C shell**, setup file in home directory: .cshrc

8.23.4 Text editors
To create, read, write, or edit text files, e.g. FLACS-CFD input file you must know how to use a text editor. Recommended text editors are:

- **vi** / **vim**
- **Emacs**
- **gedit**
- **kate**

**Warning:**
With Emacs: Remember to have an *empty line* at the end of the file. VIM adds an extra line automatically.

**Attention:**
Notepad++ is recommended for editing text files in Windows. Notepad++ does not change your text file in undesired ways (e.g., by adding/changing line feeds), which can be the case for other editors.

8.23.5 Communication with other computers

- **ssh** (SSH client) is a program for logging into a remote machine and for executing commands on a remote machine.
- **scp** copies files between hosts on a network. It uses **ssh** for data transfer.
- **ftp** is the user interface to the Internet standard File Transfer Protocol. The program allows you to transfer files to and from a remote network site.

8.23.6 Help in Linux
Most commands in Linux have related *manual* pages. These can be displayed by:

```
> man <command>
```

For instance:

```
> man ls
```

The **help** and **info** commands provide less extensive output than **man**, whereas **apropos** also includes the man output for related commands. Most commands also show help by writing

```
> <command> --help
```

For instance

```
> r1file --help
```
8.23 Linux quick reference

8.23.7 Useful commands

Recent commands are saved in a history file, located in the user's home directory, and it is listed with the command:

```
> history
```

Each user can define command aliases with `alias`. To make them available, the `alias` commands need to be executed every time a new shell is used and therefore they are usually added to the shell setup-file (.bashrc or .cshrc). `alias` can be used to make new short-cuts or to change the output of already existing commands. The `alias` syntax differs slightly from shell to shell. See:

```
> man alias
```

The `cd` command changes the working directory:

```
> cd DIR
```

To move to the parent directory:
```
> cd ..
```

`ls` lists contents of directories:
```
> ls
```

There are a lot of options to the `ls` command. See
```
> man ls
```

Other frequently used commands:

- `chgrp`: Changes group.
- `chmod`: Changes permissions.
- `chown`: Changes ownership.
- `cp`: Copies files from one place to another, or duplicates one file under a different name.
- `diff`: Compare files line by line.
- `df`: Keeps track of your hard disk space.
- `du`: Lists the file sizes in kilobyte.
- `exit`: Ends the application.
- `find`: Looks for files with particular content.
- `free`: Outputs the amount of free RAM on the system.
- `grep`: Finds words in files.
- `gunzip`: Expands files.
- `gzip`: Compresses files.
- `kill`: Terminates a process.
- `less`: Views file contents.
- `mkdir`: Creates directories.
- `more`: Views file contents.
- `mv`: Moves or renames files.
• **ps**: Lists running processes on the system.
• **pwd**: Prints the path of the working directory.
• **rm**: Deletes files permanently.
• **rmdir**: Deletes empty directories.
• **tail**: Views the last part of a file. It is an useful command for monitoring FLACS-CFD running files.
• **tar**: Assembles files into a package and extract a package.
• **top**: Provides live summary of running processes.
• **|**: Pipe command, used together with other commands.

### 8.23.8 Permissions

Permissions are best showed by an example with a directory containing only one file (`file.txt`) and one directory (`DIR`). `ls -al` gives the following results:

```
drwxr-sr-x 2 idar gexcon 4096 2008-05-17 13:01 DIR/
-rw-r--r-- 1 idar gexcon 102030 2008-05-17 14:01 file.txt
```

where:

- `-rw-r--r--` indicates the file permissions.
- `1` indicates that there is one file.
- `idar` indicates that the file belongs to the user `idar`.
- `gexcon` indicates the group.
- `102030` is the size of the file in bytes.
- `2008-05-17` indicates the date the file was created/modified/moved.
- `14:01` indicates the time the file was created/modified/moved.
- `file.txt` is the file name.

The first character in file permission is `'d'` if it is a directory and `'.` else. The next nine characters indicate the permissions, where the first three are for the user that owns the file, the next three are for the group, and the last three are for others. There are three possible attributes:

- `r`: Read permission.
- `w`: Write permission.
- `x`: Execute permission.

In the example,

```
-rw-r--r--
```

only the user has both read and write permission, i.e. can modify the file. Members of the group and others can read the file.
Chapter 9

Nomenclature

9.1 Roman letters

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Area</td>
<td>m²</td>
</tr>
<tr>
<td>α</td>
<td>Moles of O₂ in a stoichiometric reaction</td>
<td>-</td>
</tr>
<tr>
<td>c</td>
<td>Speed of sound</td>
<td>m s⁻¹</td>
</tr>
<tr>
<td>c_p</td>
<td>Specific heat capacity at constant pressure</td>
<td>J K⁻¹ kg⁻¹</td>
</tr>
<tr>
<td>c_v</td>
<td>Specific heat capacity at constant volume</td>
<td>J K⁻¹ kg⁻¹</td>
</tr>
<tr>
<td>C₁ε</td>
<td>Constant in the $k - \varepsilon$ equation; typically $C_{1\varepsilon} = 1.44$</td>
<td>-</td>
</tr>
<tr>
<td>C₂ε</td>
<td>Constant in the $k - \varepsilon$ equation; typically $C_{2\varepsilon} = 1.92$</td>
<td>-</td>
</tr>
<tr>
<td>C₃ε</td>
<td>Constant in the $k - \varepsilon$ equation; typically $C_{3\varepsilon} = 0.8$</td>
<td>-</td>
</tr>
<tr>
<td>C_D</td>
<td>Drag coefficient</td>
<td>-</td>
</tr>
<tr>
<td>C_µ</td>
<td>Constant in the $k - \varepsilon$ equation; typically $C_\mu = 0.09$</td>
<td>-</td>
</tr>
<tr>
<td>d</td>
<td>Diameter</td>
<td>m</td>
</tr>
<tr>
<td>D</td>
<td>Diffusion coefficient</td>
<td>m² s⁻¹</td>
</tr>
<tr>
<td>f</td>
<td>Sub-grid obstructions turbulence generation factor</td>
<td>-</td>
</tr>
<tr>
<td>E⁺</td>
<td>Constant in wall functions; typically $E^+ = 11$</td>
<td>-</td>
</tr>
<tr>
<td>F</td>
<td>Force</td>
<td>N</td>
</tr>
<tr>
<td>F_D</td>
<td>Drag force</td>
<td>N</td>
</tr>
<tr>
<td>F_g</td>
<td>Specific gravity force (Pool model)</td>
<td>N kg⁻¹</td>
</tr>
<tr>
<td>F_w</td>
<td>Wall friction force</td>
<td>N</td>
</tr>
<tr>
<td>F/O</td>
<td>Fuel-oxidant ratio, see definition</td>
<td>-</td>
</tr>
<tr>
<td>F_τ</td>
<td>Specific friction force (Pool model)</td>
<td>N kg⁻¹</td>
</tr>
<tr>
<td>g, \vec{g}</td>
<td>Gravitational acceleration (scalar, vector)</td>
<td>m s⁻²</td>
</tr>
<tr>
<td>h</td>
<td>Specific enthalpy</td>
<td>J kg⁻¹</td>
</tr>
<tr>
<td>h</td>
<td>Heat transfer coefficient</td>
<td>W K⁻¹ m⁻²</td>
</tr>
<tr>
<td>h</td>
<td>Height of the atmospheric mixing layer</td>
<td>m</td>
</tr>
<tr>
<td>h</td>
<td>Pool height</td>
<td>m</td>
</tr>
<tr>
<td>I_p</td>
<td>Pressure impulse</td>
<td>Pas</td>
</tr>
<tr>
<td>I_T</td>
<td>Relative turbulence intensity</td>
<td>-</td>
</tr>
<tr>
<td>k</td>
<td>Turbulent kinetic energy</td>
<td>m² s⁻²</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Unit</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------</td>
</tr>
<tr>
<td>$L$</td>
<td>Monin-Obukhov length scale</td>
<td>m</td>
</tr>
<tr>
<td>$l$</td>
<td>Length</td>
<td>m</td>
</tr>
<tr>
<td>$l_{LT}$</td>
<td>Mixing length in the $\beta$-model, $l_{LT} = C_\mu k^{3/2} \varepsilon^{-1}$</td>
<td>m</td>
</tr>
<tr>
<td>$M, M_k$</td>
<td>Molecular weight of a mixture, species</td>
<td>kg mol$^{-1}$</td>
</tr>
<tr>
<td>$m$</td>
<td>Mass</td>
<td>kg</td>
</tr>
<tr>
<td>$\dot{m}$</td>
<td>Mass rate</td>
<td>kg s$^{-1}$</td>
</tr>
<tr>
<td>$\dot{m}_L$</td>
<td>Mass leakage rate</td>
<td>kg m$^{-2}$ s$^{-1}$</td>
</tr>
<tr>
<td>$\dot{m}_V$</td>
<td>Mass evaporation rate</td>
<td>kg m$^{-2}$ s$^{-1}$</td>
</tr>
<tr>
<td>$N$</td>
<td>Total number density</td>
<td>-</td>
</tr>
<tr>
<td>$n$</td>
<td>Number density</td>
<td>-</td>
</tr>
<tr>
<td>$p$</td>
<td>Absolute pressure</td>
<td>Pa</td>
</tr>
<tr>
<td>$p_0$</td>
<td>Ambient pressure</td>
<td>Pa</td>
</tr>
<tr>
<td>$p_g$</td>
<td>Partial pressure of a gas</td>
<td>Pa</td>
</tr>
<tr>
<td>$P$</td>
<td>Gauge pressure, overpressure</td>
<td>Pa, bar</td>
</tr>
<tr>
<td>$Q$</td>
<td>Heat</td>
<td>J</td>
</tr>
<tr>
<td>$\dot{Q}$</td>
<td>Heat flow rate</td>
<td>J s$^{-1}$</td>
</tr>
<tr>
<td>$\dot{q}$</td>
<td>Specific heat rate</td>
<td>J m$^{-2}$ s$^{-1}$</td>
</tr>
<tr>
<td>$R, R_k$</td>
<td>Gas constant of a mixture, species $R = R_u/M$</td>
<td>J kg$^{-1}$ K$^{-1}$</td>
</tr>
<tr>
<td>$R_u$</td>
<td>Universal gas constant</td>
<td>8.314 J mol$^{-1}$ K$^{-1}$</td>
</tr>
<tr>
<td>$R_{\text{fuel}}$</td>
<td>Reaction rate for fuel</td>
<td>kg m$^{-3}$ s$^{-1}$</td>
</tr>
<tr>
<td>$r$</td>
<td>Radius</td>
<td>m</td>
</tr>
<tr>
<td>$r_{\text{ox}}$</td>
<td>Stoichiometric fuel-oxidant ratio on mass basis</td>
<td>-</td>
</tr>
<tr>
<td>$r_{\text{air}}$</td>
<td>Stoichiometric fuel-air ratio on mass basis</td>
<td>-</td>
</tr>
<tr>
<td>$S_L$</td>
<td>Laminar burning velocity</td>
<td>m s$^{-1}$</td>
</tr>
<tr>
<td>$S_{QL}$</td>
<td>Quasi-laminar burning velocity</td>
<td>m s$^{-1}$</td>
</tr>
<tr>
<td>$S_T$</td>
<td>Turbulent burning velocity</td>
<td>m s$^{-1}$</td>
</tr>
<tr>
<td>$T$</td>
<td>Absolute temperature</td>
<td>K</td>
</tr>
<tr>
<td>$T^+$</td>
<td>Dimensionless temperature in wall functions</td>
<td>-</td>
</tr>
<tr>
<td>$t$</td>
<td>Time</td>
<td>s</td>
</tr>
<tr>
<td>$U_0$</td>
<td>Reference, characteristic velocity</td>
<td>m s$^{-1}$</td>
</tr>
<tr>
<td>$u_i, \vec{u}$</td>
<td>Mean velocity ($i$th component, vector)</td>
<td>m s$^{-1}$</td>
</tr>
<tr>
<td>$u'$</td>
<td>Root mean square of velocity</td>
<td>m s$^{-1}$</td>
</tr>
<tr>
<td>$u^*$</td>
<td>Friction velocity</td>
<td>m s$^{-1}$</td>
</tr>
<tr>
<td>$V$</td>
<td>Volume</td>
<td>m$^3$</td>
</tr>
<tr>
<td>$\dot{V}$</td>
<td>Volume rate</td>
<td>m$^3$ s$^{-1}$</td>
</tr>
<tr>
<td>$W^*$</td>
<td>Dimensionless reaction rate</td>
<td>-</td>
</tr>
<tr>
<td>$X$</td>
<td>Mole fraction</td>
<td>-</td>
</tr>
<tr>
<td>$x$</td>
<td>Length coordinate</td>
<td>m</td>
</tr>
<tr>
<td>$x$</td>
<td>Concentration of gas, $x = p_g/p_0$</td>
<td>mol/mol</td>
</tr>
<tr>
<td>$x^+$</td>
<td>Dimensionless concentration in wall functions</td>
<td>-</td>
</tr>
<tr>
<td>$Y$</td>
<td>Mass fraction</td>
<td>-</td>
</tr>
<tr>
<td>$y$</td>
<td>Wall distance</td>
<td>m</td>
</tr>
</tbody>
</table>
9.3 Subscripts

| $y^+$ | Dimensionless wall distance in wall functions | - |
| $z$ | Distance above the ground | m |
| $z_0$ | Aerodynamical roughness length | m |

9.2 Greek letters

| $\alpha$ | Volume fraction | - |
| $\alpha$ | Thermal diffusivity | m$^2$ s$^{-1}$ |
| $\beta$ | Transformation factor in the $\beta$-model, see The FLACS-CFD flame model | - |
| $\beta_i$ | Area porosity in the $i$th direction | - |
| $\beta_v$ | Volume porosity | - |
| $\gamma$ | Isentropic ratio | - |
| $\gamma_p$ | Pressure exponent for the laminar burning velocity, see correlation. | - |
| $\Delta$ | Control volume length | m |
| $\Delta h_{fg}$ | Heat of evaporation | J kg$^{-1}$ |
| $\delta_H$ | Heaviside step function. $\delta_H(a - b) = 1$ if $a \geq b$. $\delta_H(a - b) = 0$ if $a < b$. | - |
| $\delta_{ij}$ | Kronecker delta function. $\delta_{ij} = 1$ if $i = j$. $\delta_{ij} = 0$ if $i \neq j$. | - |
| $\varepsilon$ | Dissipation of turbulent kinetic energy | m$^2$ s$^{-3}$ |
| $\varepsilon_g$ | Surface roughness | m |
| $\zeta$ | Surface tension | N m$^{-1}$ |
| $\theta$ | Specific enthalpy in pool model. | J kg$^{-1}$ |
| $\kappa$ | Von Karman constant; typically $\kappa = 0.41$. | - |
| $\lambda$ | Conductivity | W m$^{-1}$ K$^{-1}$ |
| $\lambda$ | Detonation cell size | m |
| $\mu$ | Dynamic viscosity | Pa s |
| $\mu_t$ | Dynamic turbulent viscosity | Pa s |
| $\mu_{eff}$ | Effective viscosity, $\mu_{eff} = \mu + \mu_t$ | Pa s |
| $\nu$ | Kinematic viscosity | m$^2$ s$^{-1}$ |
| $\xi$ | Mixture fraction | - |
| $\rho$ | Density | kg m$^{-3}$ |
| $\sigma$ | Prandtl-Schmidt number, see overview. | - |
| $\sigma_{ij}$ | Stress tensor, see equation. | N m$^{-2}$ |
| $\tau$ | Time scale | s |
| $\tau_e$ | Integral time scale in turbulent flows | s |
| $\tau_w$ | Wall shear stress | N m$^{-2}$ |
| $\Phi$ | Equivalence ratio, see definition. | - |
| $\phi$ | General variable | - |
| $\chi$ | Progress variable, see definition. | - |

9.3 Subscripts

| $a$ | Ambient | - |
### Nomenclature

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$cv$</td>
<td>Control volume</td>
<td>-</td>
</tr>
<tr>
<td>$D$</td>
<td>Drag</td>
<td>-</td>
</tr>
<tr>
<td>$g$</td>
<td>Ground</td>
<td>-</td>
</tr>
<tr>
<td>$f$</td>
<td>Flow</td>
<td>-</td>
</tr>
<tr>
<td>$i$</td>
<td>Species index, spatial index</td>
<td>-</td>
</tr>
<tr>
<td>$j$</td>
<td>Spatial index</td>
<td>-</td>
</tr>
<tr>
<td>$L$</td>
<td>Laminar</td>
<td>-</td>
</tr>
<tr>
<td>$n$</td>
<td>Control volume index</td>
<td>-</td>
</tr>
<tr>
<td>$o$</td>
<td>Sub-grid objects</td>
<td>-</td>
</tr>
<tr>
<td>$p$</td>
<td>Particle</td>
<td>-</td>
</tr>
<tr>
<td>stoich</td>
<td>Stochiometric</td>
<td>-</td>
</tr>
<tr>
<td>$T$</td>
<td>Turbulent</td>
<td>-</td>
</tr>
<tr>
<td>$v$</td>
<td>Volume</td>
<td>-</td>
</tr>
<tr>
<td>$w$</td>
<td>Wall</td>
<td>-</td>
</tr>
</tbody>
</table>

### 9.4 Dimensionless groups

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Bi$</td>
<td>Biot number, $Bi = \frac{hd_p}{\lambda p}$</td>
<td>-</td>
</tr>
<tr>
<td>$Da$</td>
<td>Damköhler number, $Da = \frac{\tau_{chem}}{\tau}$</td>
<td>-</td>
</tr>
<tr>
<td>$Fr$</td>
<td>Froude number, $Fr = \frac{\text{inertial force}}{\text{gravity force}} = \frac{u}{\sqrt{g}}$</td>
<td>-</td>
</tr>
<tr>
<td>$Le$</td>
<td>Lewis number, $Le = \frac{\tau p c_p D}{\nu} = \frac{Sc}{Pr}$</td>
<td>-</td>
</tr>
<tr>
<td>$Ma$</td>
<td>Mach number, $Ma = \frac{u}{c}$</td>
<td>-</td>
</tr>
<tr>
<td>$Nu$</td>
<td>Nusselt number, $Nu = \frac{hl}{\lambda}$</td>
<td>-</td>
</tr>
<tr>
<td>$Pr$</td>
<td>Prandtl number, $Pr = \frac{\nu}{\alpha} = \frac{\mu c_p}{\lambda}$</td>
<td>-</td>
</tr>
<tr>
<td>$Re$</td>
<td>Reynolds number, $Re = \frac{\text{inertial force}}{\text{viscous force}} = \frac{\rho u l}{\mu}$</td>
<td>-</td>
</tr>
<tr>
<td>$Sc$</td>
<td>Schmidt number, $Sc = \frac{\nu}{D} = \frac{\mu}{\rho D}$</td>
<td>-</td>
</tr>
<tr>
<td>$St$</td>
<td>Stokes number, $St = \frac{\tau_p}{\tau_f}$</td>
<td>-</td>
</tr>
<tr>
<td>$We$</td>
<td>Weber number, $We = \frac{\text{inertial force}}{\text{surface tension force}} = \frac{\rho u^2 d_p}{\zeta}$</td>
<td>-</td>
</tr>
</tbody>
</table>

### 9.5 Abbreviations

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIT</td>
<td>Auto Ignition Temperature</td>
<td>-</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer Aided Design</td>
<td>-</td>
</tr>
<tr>
<td>CASD</td>
<td>Computer Aided Scenario Design</td>
<td>-</td>
</tr>
<tr>
<td>CFD</td>
<td>Computational Fluid Dynamics</td>
<td>-</td>
</tr>
<tr>
<td>CFL</td>
<td>Courant-Friedrichs-Levy</td>
<td>-</td>
</tr>
<tr>
<td>CMR</td>
<td>Christian Michelsen Research</td>
<td>-</td>
</tr>
<tr>
<td>CP8</td>
<td>Convex Polyhedron</td>
<td>-</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
<td>-</td>
</tr>
<tr>
<td>CV</td>
<td>Control Volume</td>
<td>-</td>
</tr>
<tr>
<td>DDT</td>
<td>Deflagration-to-Detonation Transition</td>
<td>-</td>
</tr>
</tbody>
</table>
### 9.6 FLACS-CFD variables

| DESC  | Dust Explosion Simulation Code | - |
| DNS   | Direct Numerical Simulation    | - |
| ER    | Equivalence Ratio             | - |
| FLACS | FLame ACceleration Simulator  | - |
| GTC   | General Truncated Cone        | - |
| HSL   | Health and Safety Laboratory  | - |
| HVAC  | Heating, Ventilating, and Air-Conditioning | - |
| LES   | Large Eddy Simulation         | - |
| LFL   | Lower Flammability Limit      | - |
| LSLIM | Characteristic Geometrical Dimension | - |
| MIE   | Minimum Ignition Energy       | - |
| PI    | Ignition Probability          | - |
| QRA   | Quantitative Risk Analyses    | - |
| RAM   | Random Access Memory          | - |
| UFL   | Upper Flammability Limit      | - |
| CGNS  | CFD General Notation System   | - |
| SIDS  | Standard Interface Data Structures | - |
| PBR   | Physically based rendering    | - |

#### 9.6.1 FLACS-CFD variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>CFLC</td>
<td>CFL number based on speed of sound</td>
<td>-</td>
</tr>
<tr>
<td>CFLV</td>
<td>CFL number based on flow velocity</td>
<td>-</td>
</tr>
<tr>
<td>CS</td>
<td>Speed of sound, (c)</td>
<td>(\text{m s}^{-1})</td>
</tr>
<tr>
<td>DPDT</td>
<td>Rate of pressure rise, (\frac{dp}{dt})</td>
<td>(\text{Pa s}^{-1})</td>
</tr>
<tr>
<td>EPK</td>
<td>Turbulence ratio, (\varepsilon)</td>
<td>(\text{s}^{-1})</td>
</tr>
<tr>
<td>EPS</td>
<td>Dissipation of turbulent kinetic energy, (\varepsilon)</td>
<td>(\text{m}^2 \text{s}^{-2})</td>
</tr>
<tr>
<td>EQ</td>
<td>Equivalence ratio, finite bounded, (\frac{(F/O)}{(F/O)_{\text{stoich}}})</td>
<td>-</td>
</tr>
<tr>
<td>ER</td>
<td>Equivalence ratio, (\Phi)</td>
<td>-</td>
</tr>
<tr>
<td>DRAG_IMP</td>
<td>Drag impulse, (\int \left</td>
<td>\vec{F}_D \right</td>
</tr>
<tr>
<td>DRAG</td>
<td>Drag value, (\left</td>
<td>\vec{F}_D \right</td>
</tr>
<tr>
<td>FDOSE</td>
<td>Dose, integral of mole fraction of fuel, (\int X_{\text{fuel}} dt)</td>
<td>(\text{s})</td>
</tr>
<tr>
<td>FLUX</td>
<td>Mass flux, (\frac{\dot{m}}{A})</td>
<td>(\text{kg m}^{-2} \text{s}^{-1})</td>
</tr>
<tr>
<td>FMIX</td>
<td>Mixture fraction, (\xi)</td>
<td>-</td>
</tr>
<tr>
<td>FMOLE</td>
<td>Mole fraction of fuel, (X_{\text{fuel}})</td>
<td>-</td>
</tr>
<tr>
<td>FUEL</td>
<td>Mass fraction of fuel, (Y_{\text{fuel}})</td>
<td>-</td>
</tr>
<tr>
<td>GAMMA</td>
<td>Isentropic gas constant, (\gamma)</td>
<td>-</td>
</tr>
<tr>
<td>H</td>
<td>Enthalpy, (h)</td>
<td>(\text{J kg}^{-1})</td>
</tr>
<tr>
<td>K</td>
<td>Turbulent kinetic energy, (k)</td>
<td>(\text{J kg}^{-1})</td>
</tr>
<tr>
<td>LT</td>
<td>Turbulence length scale, (output), (l_{LT})</td>
<td>(\text{m})</td>
</tr>
<tr>
<td>MACH</td>
<td>Mach number, (Ma)</td>
<td>-</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td>Unit</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td><strong>MU</strong></td>
<td>Effective dynamic viscosity, $\mu_{\text{eff}}$</td>
<td>Pa·s</td>
</tr>
<tr>
<td><strong>NUSSN</strong></td>
<td>Nusselt number, Nu</td>
<td>-</td>
</tr>
<tr>
<td><strong>OX</strong></td>
<td>Mass fraction of oxygen, $Y_{O_2}$</td>
<td>-</td>
</tr>
<tr>
<td><strong>P</strong></td>
<td>Gauge pressure, overpressure, $P$</td>
<td>bar</td>
</tr>
<tr>
<td><strong>P_{\text{IMP}}</strong></td>
<td>Pressure impulse, $\int_{t_1}^{t_2} P dt$</td>
<td>bar·s</td>
</tr>
<tr>
<td><strong>P_{\text{MAX}}</strong></td>
<td>Maximum over pressure, $P_{\text{max}}$</td>
<td>bar</td>
</tr>
<tr>
<td><strong>PROD</strong></td>
<td>Mass fraction of products, $Y_{\text{prod}}$</td>
<td>-</td>
</tr>
<tr>
<td><strong>RET</strong></td>
<td>Turbulent Reynolds number, $\frac{\mu_{\text{eff}}}{\mu_{\text{laminar}}}$</td>
<td>-</td>
</tr>
<tr>
<td><strong>RFU</strong></td>
<td>Combustion rate, $R_{\text{fuel}}$</td>
<td>kg·m$^{-3}$·s$^{-1}$</td>
</tr>
<tr>
<td><strong>RHO</strong></td>
<td>Density, $\rho$</td>
<td>kg·m$^{-3}$</td>
</tr>
<tr>
<td><strong>RTI</strong></td>
<td>Relative turbulence intensity (input), $I_T$</td>
<td>-</td>
</tr>
<tr>
<td><strong>T</strong></td>
<td>Temperature, $T$</td>
<td>K</td>
</tr>
<tr>
<td><strong>TAUWX</strong></td>
<td>Wall shear stress in x direction, $\tau_{w,1}$</td>
<td>N·m$^{-2}$</td>
</tr>
<tr>
<td><strong>TAUWY</strong></td>
<td>Wall shear stress in y direction, $\tau_{w,2}$</td>
<td>N·m$^{-2}$</td>
</tr>
<tr>
<td><strong>TAUWZ</strong></td>
<td>Wall shear stress in z direction, $\tau_{w,3}$</td>
<td>N·m$^{-2}$</td>
</tr>
<tr>
<td><strong>TLS</strong></td>
<td>Turbulence length scale (input), $l_{LT}$</td>
<td>m</td>
</tr>
<tr>
<td><strong>TURB</strong></td>
<td>Root mean square of velocity, $u'$</td>
<td>m$^{-1}$</td>
</tr>
<tr>
<td><strong>TURBI</strong></td>
<td>Relative turbulence intensity (output), $I_T$</td>
<td>-</td>
</tr>
<tr>
<td><strong>U</strong></td>
<td>Velocity component in x direction, $u_1$</td>
<td>m$^{-1}$</td>
</tr>
<tr>
<td><strong>UDRAG_{\text{IMP}}</strong></td>
<td>Drag impulse in x direction, $\int_0^t F_{D,1} dt$</td>
<td>Pa·s</td>
</tr>
<tr>
<td><strong>UDRAG</strong></td>
<td>Drag value in x direction, $</td>
<td>F_{D,1}</td>
</tr>
<tr>
<td><strong>UFLUX</strong></td>
<td>Mass flux in x direction, $\rho u_1$</td>
<td>kg·m$^{-2}$·s$^{-1}$</td>
</tr>
<tr>
<td><strong>UVW</strong></td>
<td>Absolute value of velocity, $</td>
<td>\vec{u}</td>
</tr>
<tr>
<td><strong>V</strong></td>
<td>Velocity component in y direction, $u_2$</td>
<td>m$^{-1}$</td>
</tr>
<tr>
<td><strong>VDRAG_{\text{IMP}}</strong></td>
<td>Drag impulse in y direction, $\int_0^t F_{D,2} dt$</td>
<td>Pa·s</td>
</tr>
<tr>
<td><strong>VDRAG</strong></td>
<td>Drag value in y direction, $</td>
<td>F_{D,2}</td>
</tr>
<tr>
<td><strong>VFLUX</strong></td>
<td>Mass flux in y direction, $\rho u_2$</td>
<td>kg·m$^{-2}$·s$^{-1}$</td>
</tr>
<tr>
<td><strong>VVEC</strong></td>
<td>Velocity vector, $\vec{u}$</td>
<td>m$^{-1}$</td>
</tr>
<tr>
<td><strong>W</strong></td>
<td>Velocity component in z direction, $u_3$</td>
<td>m$^{-1}$</td>
</tr>
<tr>
<td><strong>WDRAG_{\text{IMP}}</strong></td>
<td>Drag impulse in z direction, $\int_0^t F_{D,3} dt$</td>
<td>Pa·s</td>
</tr>
<tr>
<td><strong>WDRAG</strong></td>
<td>Drag value in z direction, $</td>
<td>F_{D,3}</td>
</tr>
<tr>
<td><strong>WFLUX</strong></td>
<td>Mass flux in z direction, $\rho u_3$</td>
<td>kg·m$^{-2}$·s$^{-1}$</td>
</tr>
<tr>
<td><strong>DPDX</strong></td>
<td>Spatial Pressure Gradient (Normalized)</td>
<td>-</td>
</tr>
<tr>
<td><strong>DDTLS</strong></td>
<td>Detonation Length Scale, $\frac{LSLIM}{\lambda}$</td>
<td>-</td>
</tr>
<tr>
<td><strong>ACH</strong></td>
<td>Air change rate per hour (only accessible in Flowvis and only when</td>
<td>l/h</td>
</tr>
<tr>
<td></td>
<td>VVEC is output)</td>
<td></td>
</tr>
<tr>
<td><strong>FAT</strong></td>
<td>Flame arrival time</td>
<td>ms</td>
</tr>
</tbody>
</table>
Chapter 10

Validation

10.1 Introduction

Validation against analytical solutions, experiments, and real incidents is the preferred way of documenting the range of scenarios for which a model system generates reliable estimates for relevant physical phenomena. Systematic validation is also essential for quality assurance: continuously monitoring the effects of changes in source code, compilers, operating systems. As such, Gexcon has implemented an automated framework for validation of the FLACS-CFD software. Currently, there are approximately 60 validation series included in the automated framework, with approximately 2,700 FLACS-CFD simulations (including a range sensitivity simulations). The cases range from small scale basic verification cases to large scale dispersion and explosion cases.

In this section, a small subset of the validation cases is presented as Validation Summary Sheets to highlight the validation status of FLACS-CFD.

10.2 Structure of Validation Summary Sheets

The validation results are presented as succinct validation documents (Validation Summary Sheets) that clearly communicate the FLACS-CFD performance for particular test cases.

10.2.1 Experiment details

The first page of the Validation Summary Sheet is used primarily to present the details of the experiment, highlighting the relevance for FLACS-CFD validation.

10.2.2 FLACS-CFD simulation results

10.2.2.1 Performance metric parameters

Five performance metrics are provided for each campaign. These are

- the geometric mean bias (MG),

\[
MG = \exp(\ln X_p - \ln X_o) \tag{10.1}
\]

- the geometric variance (VG),

\[
VG = \exp[(\ln X_p - \ln X_o)^2] \tag{10.2}
\]

- the fraction of predictions within a factor of two of observations (FAC2),

\[
0.5 \leq \frac{X_p}{X_o} \leq 2.0 \tag{10.3}
\]
• the fractional bias (FB), and

\[
FB = \frac{\bar{X}_p - \bar{X}_o}{0.5(\bar{X}_p + \bar{X}_o)}
\]  \hspace{1cm} (10.4)

• the normalized mean square error (NMSE).

\[
NMSE = \frac{(X_p - X_o)^2}{X_p X_o}
\]  \hspace{1cm} (10.5)

where \(X_p\) are the model predictions, \(X_o\) and the observations and the overbar (e.g. \(\bar{X}_o\)) is the average over the dataset. The number of data points used to calculate each metric (N) is also shown. Bounds are provided for each performance metric to show how they vary across the grid sizes considered for each validation exercise.

A perfect model would have MG, VG, and FAC2 = 1.0; and FB and NMSE = 0.0. Of course, there is no such thing as a perfect model. The individual performance metrics are coloured green if they fall in what is generally considered the acceptable range, i.e.:

• The fraction of predictions within a factor of two of observations is at least 50% (i.e. FAC2 > 0.5).

• The mean bias is within a factor of 2 of the mean (i.e. \(-0.67 < FB < 0.67\) or \(0.5 < MG < 2\)).

• The random scatter is about a factor of two to three of the mean (i.e. \(NMSE < 1.5\) or \(VG < 4\)).

Note the above ranges are indicative only. Multiple performance measures should be applied and considered in any model evaluation exercise, as each measure has advantages and disadvantages and there is not a single measure that is universally applicable to all conditions.

**10.2.2.2 Performance plots**

Two plots types are shown to aid evaluation of model performance; parabola plots and scatter plots. Figures 10.1 and 10.2 present some examples of the scatter plots and their corresponding parabola plots.

**10.3 Validation cases**

Two-pagers for 50 experimental campaigns are presented, including two new cases for pool fires. A list of the cases is given under each subsection.

**10.3.1 Discussion**

All validation cases have been run with grids within the grid guidelines. Refer to section “Recommended grid configuration” for discussion about the grid guidelines and a grid sensitivity assessment. To represent the much more rapid progression of detonation and have a consistent approach across both deflagration, DDT and detonation cases, we have reduced the averaging time of pressure to 0.1ms for the experimental results in all explosion cases (from 1.5ms previously). For deflagrations this is more conservative and may result in slightly increased underprediction in some of the validation cases, as the typical timestep in FLACS-CFD simulations will be longer than this. As the applied averaging time is different, validation performance cannot be exactly compared with previous validation, although in most cases the difference will be modest.
10.3 Validation cases

(a) Scatter plot, $X_p = X_o$

(b) Parabola plot, $X_p = X_o$

(c) Scatter plot, $X_p > X_o$

(d) Parabola plot, $X_p > X_o$

(e) Scatter plot, $X_p < X_o$

(f) Parabola plot, $X_p < X_o$

Figure 10.1: Examples of idealised data variation (low scatter) presented in scatter and parabola plots.
(a) Scatter plot, $X_p \neq X_o$  
(b) Parabola plot, $X_p \neq X_o$

Figure 10.2: Example of data variation (high scatter) presented in scatter and parabola plots.

Figure 10.3: Scatter plot of validation cases: Explosion (top-left), Dispersion (top-centre) and Fire (top-right).
Table 10.1: Gas explosion validation cases.

10.3.2 Gas Explosion

Figure 10.3 (top-left) summarizes the presented explosion validation cases in a single scatter plot, while table 10.1 gives a summary of the individual experiments. Performance is generally good, though we do see a spread in the results. There is a tendency for FLACS-CFD to perform better for large-scale experiments. This is because experiments at smaller scales are more sensitive to the modelling of quasi-laminar flame propagation and to radiative and convective heat losses. Different mixtures are included in the validation and the performance across different gas species is generally good. However, there are some cases of poor representation of very fast flames; for example, propane and ethylene overpressures are under-predicted. The under-prediction is especially evident for the ethylene cases. FLACS-CFD does not capture the flame instability types that become important at very high flame-speeds (approaching DDT), so the deviations in the results for propane and ethylene are somewhat expected. In the performance plots, detonation cases are distinguished using a different marker style, as shown in the plot legend. Those are the hydrogen-explosion tests for which detonation was observed in the experiment. In the validation, FLACS-CFD is run with default settings for detonation (DDT "AUTO" and DDT_TIME -1), meaning that detonation is triggered for hydrogen explosions only when DPDX exceeds 1. In addition to the validation against experimental data, the result of which is presented in the following subsections, FLACS-CFD version 22.2 has been tested for hydrogen explosion using a set of realistic full-scale scenarios. Simulations indicate a higher frequency of detonation events compared to the detonation frequency obtained in the simulation of hydrogen explosion validation cases. For hydrogen-explosion, thanks to the new detonation model in FLACS-CFD version 22.2, on average the user should expect an increase of predicted peak pressure of about 20% compared to the peak pressure obtained from simulating the same scenarios with version 22.1.
Small scale hydrogen explosion experiments were performed in 3D-corner tests (an obstacle array of 37x37x37cm that is blocked on 3 sides and open on 3 sides). The tests considered 3 different obstruction densities, 2 ignition locations, and a range of H2 concentrations.

Location: Gexcon AS (NORWAY)
Year: 2004
Number of cases in series: 32

Available measurements: Pressure

Grids used: 23 mm

REFERENCES
### POSTPROCESS QUANTITIES

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>87</td>
<td>2.7</td>
<td>6.1</td>
<td>0.31</td>
<td>0.59</td>
<td>1.6</td>
</tr>
<tr>
<td>HYDROGEN, N2</td>
<td>36</td>
<td>2.9</td>
<td>7.7</td>
<td>0.33</td>
<td>0.64</td>
<td>2</td>
</tr>
<tr>
<td>All fuels</td>
<td>123</td>
<td>2.8</td>
<td>6.6</td>
<td>0.32</td>
<td>0.6</td>
<td>1.7</td>
</tr>
</tbody>
</table>

#### Validation cases

- **10.3 Validation cases**

---

**FLACS-CFD 22.2 User’s Manual**
### BAKERRISK

**HYDROGEN, ETHYLENE**

A set of large-scale unconfined vapor cloud explosion tests were performed in a congested region. The test rig consisted of cubic sections with 1.8 m long tubes located vertically serving as obstacles. Ethylene-air mixtures with concentration varying from lean to rich and lean hydrogen-air mixtures were tested. The mixtures were ignited at the centre of rig just above the ground level. Pressure measurements were made using arrays of pressure gauges placed up to 91 m away from the rig.

<table>
<thead>
<tr>
<th>Location:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Year:</td>
<td></td>
</tr>
<tr>
<td>Number of cases in series:</td>
<td>4</td>
</tr>
</tbody>
</table>

### SELECTION OF TESTS

Available measurements: Pressure

### SIMULATION SETUP

Grids used: 120 mm

### REFERENCES

Kelly Thomas, Quentin Baker, Don Ketchum, Martin Goodrich, Max Kolbe (2003), Deflagration to Detonation Transition in Unconfined Vapor Cloud Explosions
## POSTPROCESS QUANTITIES

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>9</td>
<td>12</td>
<td>2</td>
<td>0.57</td>
<td>-0.065</td>
<td>0.31</td>
</tr>
<tr>
<td>ETHYLENE</td>
<td>1</td>
<td>5.2</td>
<td>15</td>
<td>0</td>
<td>1.4</td>
<td>3.4</td>
</tr>
<tr>
<td>All fuels</td>
<td>10</td>
<td>1.4</td>
<td>2.4</td>
<td>0.6</td>
<td>-0.063</td>
<td>0.35</td>
</tr>
</tbody>
</table>

![Graph showing Geometric variance vs Geometric mean bias](image1)

![Graph showing Simulation vs Experiment](image2)
10.3.2.3 BFETS Phase 2B

**BFETS P2B**

ETHANE, METHANE, PROPANE

Phase 2B of the Blast and Fire Engineering Project for Topside Structures (BFETS) considered large-scale experiments in a test vessel with an internal volume of 50 m³. The vessel included varying geometric congestion and confinement. Experiments were conducted to study the effect of variation in gas concentration, ignition location, and water spray.

Location: Spadeadam, United Kingdom
Year: 1994
Number of cases in series: 14

**SELECTION OF TESTS**

Available measurements: Flame arrival time, Pressure

**SIMULATION SETUP**

Grids used: 500 mm

**REFERENCES**


## Flame arrival time

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETHANE, METHANE, PROPANE</td>
<td>666</td>
<td>1.3</td>
<td>11</td>
<td>0.98</td>
<td>0.24</td>
<td>0.1</td>
</tr>
</tbody>
</table>

## Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETHANE, METHANE, PROPANE</td>
<td>370</td>
<td>0.53</td>
<td>2.6</td>
<td>0.58</td>
<td>-0.54</td>
<td>1.1</td>
</tr>
</tbody>
</table>
## 10.3.2.4 BFETS Phase 3A Deluge

**BFETS P3A DELUGE**

<table>
<thead>
<tr>
<th>ETHANE, METHANE, PROPANE, BUTANE, PENTANE</th>
</tr>
</thead>
</table>
Tests performed in realistic offshore compressor modules of dimensions 28m x 12m x 8m. The experimental program investigated the degree of congestion (equipment density), degree of confinement (vent area), ignition location, repeatability, and the effect of various water deluge layouts. This subset is only the water spray tests, excluding those two tests (tests 28 and 34) with vessel specific deluge. The deluge tests (35 & 36) do not include the scaffolding.

**Location:** United Kingdom  
**Year:** 1997 - 2000  
**Number of cases in series:** 19

### Selection of Tests

Available measurements: Pressure

### Simulation Setup

**Grids used:** 500 mm

### References


---

**504 Validation**

**504.3.2.4 BFETS Phase 3A Deluge**

**FLACS-CFD v22.2 User's Manual**

**GEXCON**
POSTPROCESS QUANTITIES

Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETHANE, METHANE, PROPANE</td>
<td>384</td>
<td>0.67</td>
<td>1.8</td>
<td>0.58</td>
<td>-0.43</td>
<td>1.8</td>
</tr>
<tr>
<td>BUTANE, ETHANE, METHANE, PENTANE, PROPANE</td>
<td>173</td>
<td>0.87</td>
<td>1.2</td>
<td>0.91</td>
<td>-0.1</td>
<td>0.16</td>
</tr>
<tr>
<td>PROPANE</td>
<td>71</td>
<td>0.52</td>
<td>1.7</td>
<td>0.66</td>
<td>-0.63</td>
<td>0.58</td>
</tr>
<tr>
<td>All fuels</td>
<td>628</td>
<td>0.7</td>
<td>1.6</td>
<td>0.74</td>
<td>-0.39</td>
<td>1.7</td>
</tr>
</tbody>
</table>
10.3.2.5  BFETS Phase 3A

**BFETS P3A**

ETHANE, METHANE, PROPANE, BUTANE, PENTANE

Tests performed in realistic offshore compressor modules of dimensions 28m x 12m x 8m. The experimental program investigated the degree of congestion (equipment density), degree of confinement (vent area), ignition location, repeatability, and the effect of various water deluge layouts.  

Location: United Kingdom  
Year: 1997 - 2000  
Number of cases in series: 9

**SELECTION OF TESTS**

Available measurements: Pressure

**SIMULATION SETUP**

Grids used: 500 mm

**REFERENCES**

### FLACS-CFD 22.2

**DATE:** FEBRUAR 15, 2023

**POSTPROCESS QUANTITIES**

<table>
<thead>
<tr>
<th>Fuel Type</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETHANE, METHANE, PROPANE</td>
<td>291</td>
<td>0.7</td>
<td>1.7</td>
<td>0.75</td>
<td>-0.45</td>
<td>1.2</td>
</tr>
<tr>
<td>BUTANE, ETHANE, METHANE, PENTANE, PROPANE</td>
<td>38</td>
<td>1</td>
<td>1.1</td>
<td>0.95</td>
<td>0.015</td>
<td>0.35</td>
</tr>
<tr>
<td><strong>All fuels</strong></td>
<td>329</td>
<td>0.73</td>
<td>1.6</td>
<td>0.77</td>
<td>-0.39</td>
<td>1.1</td>
</tr>
</tbody>
</table>

![Graph showing geometric variance vs. geometric mean bias](image)

**Graph Caption:**
- **Red:** BUTANE, ETHANE, METHANE, PENTANE, PROPANE
- **Blue:** ETHANE, METHANE, PROPANE

**Simulation vs. Experiment**
- **Red** and **Blue** markers represent different fuel types.

---

**10.3 Validation cases**
## BFETS Partial Fills

### BFETS PARTIAL FILLS

**ETHANE, METHANE, PROPANE**

A series of six explosion experiments were conducted. The explosion experiments were conducted in the full-scale explosion rig at BG Spadeadam, which is designed to be representative of an offshore module. The experiments studied explosions produced by ignition of a homogeneous mixture of natural gas and air with the volume of the natural gas and air mixture ranging between 10% and 100% of the volume of the explosion rig. The six tests were carried out using a lengthened module of 28m x12m x8m. Variations in the experiments included different gas cloud size without mitigation techniques.

**Location:** BG Spadeadam (UK)  
**Year:** 1994  
**Number of cases in series:** 6

### SELECTION OF TESTS

**Available measurements:** Flame arrival time, Pressure

### SIMULATION SETUP

**Grids used:** 533 mm

### REFERENCES

### Flam Arrival Time

<table>
<thead>
<tr>
<th>Fuel</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethane, Methane, Propane</td>
<td>374</td>
<td>1.1</td>
<td>1</td>
<td>0.99</td>
<td>0.1</td>
<td>0.032</td>
</tr>
</tbody>
</table>

### Pressure

<table>
<thead>
<tr>
<th>Fuel</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethane, Methane, Propane</td>
<td>127</td>
<td>1.4</td>
<td>1.5</td>
<td>0.65</td>
<td>0.19</td>
<td>0.51</td>
</tr>
</tbody>
</table>
10.3.2.7 DNVGL 180m3

DNVGL 180M3

BUTANE, ETHANE, METHANE, PENTANE, PROPANE

Large-scale vented confined explosions conducted at the Spadeadam Test Site to study the effect of vent size and congestion on vented explosions. Thirty-eight stoichiometric natural gas explosions were carried out in 182 m³ vented explosion chamber. Polyethylene pipes were used to provide congestion with volume blockage ranging from 0% to 5%.

Location: Spadeadam test site, UK
Year: 1991
Number of cases in series: 25

SELECTION OF TESTS

Available measurements: Pressure

SIMULATION SETUP

Grids used: 225 mm, 300 mm

REFERENCES

### Pressure Postprocessing Quantities

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUTANE, ETHANE, METHANE, PENTANE, PROPAPE</td>
<td>53</td>
<td>0.98 – 1.4</td>
<td>1.3 – 1.5</td>
<td>0.77 – 0.83</td>
<td>0.16 – 0.19</td>
<td>0.14 – 0.53</td>
</tr>
</tbody>
</table>

#### Graphs

- **Geometric Mean Bias vs. Geometric Variance**
- **Experiment vs. Simulation**
The project EMERGE (Extended Modelling and Experimental Research into Gas Explosions) provided a set of experimental explosion data at different scales, for two fuel types, with different obstruction parameters and with turbulence present in the flammable mixture at ignition. The geometry was similar to the MERGE project (the experiments of involved a single array of pipes).

**Location:** British Gas Fauld and Spadendam test facilities (UK)
**Year:** 1996
**Number of cases in series:** 12

### Selection of Tests

**Available measurements:** Pressure

### Simulation Setup

**Grids used:** 150 mm, 300 mm

### References

### Pressure

<table>
<thead>
<tr>
<th>Fuel</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methane</td>
<td>18</td>
<td>0.74 - 0.89</td>
<td>1 - 1.1</td>
<td>1</td>
<td>-0.3 - -0.16</td>
<td>0.049 - 0.098</td>
</tr>
<tr>
<td>Propane</td>
<td>30</td>
<td>0.62</td>
<td>1.4</td>
<td>0.77</td>
<td>-0.68</td>
<td>1.3</td>
</tr>
<tr>
<td>All fuels</td>
<td>18</td>
<td>0.74 - 0.76</td>
<td>1.1 - 1.2</td>
<td>0.89 - 1</td>
<td>-0.46 - -0.3</td>
<td>0.098 - 0.92</td>
</tr>
</tbody>
</table>

---

**10.3 Validation cases**

![Graph](image)

**FLACS-CFD v22.2 User’s Manual**
10.3.2.9 Fh-ICT Lane Experiment

The test setup consisted of a driver section that was a rectangular container (3 x 1.5 x 1.5 m). There was a square-spaced opening in the front wall of the driver section with a blocking ratio of 0.1 (tests IA1, IA2, IA3) and 0.3 (tests IA4 and IA5). The container was followed by a “lane” which consisted of two parallel walls (thickness 4 cm) at a distance of 3 m with a length of 12 m and a height of 3 m. The walls were fixed with steel braces to ensure that they do not move during the experiments. The whole volume was filled with H2-air mixture. The hydrogen mixture was enclosed within the lane using a polyethylene foil (thickness 0.2 mm). The mixture was ignited at the rear side of the container with five distributed pyrotechnic igniters. Two vertical tubes (diameter 14 cm) were installed at a distance of 5 cm from the wall each in the middle and at the end of the lane for all tests except test IA1.

Location: Fraunhofer Institute for Propellants and Explosives (ICT), Germany
Year: 1984
Number of cases in series: 4

Available measurements: Pressure

Grids used: 100 mm

REFERENCES

514 Validation
10.3.2.9 Fh-ICT Lane Experiment
FLACS-CFD v22.2 User’s Manual
10.3.2.10 Fireseal Pipe Penetration Blast Wall

**FIRESEAL PIPE PENETRATION BLAST WALL**

**PROPANE, METHANE**

Large-scale tests explosion tests were performed to study the behaviour and explosion blast resistance properties of passive fire protection from FireSeal AB. Standard Gexcon 50m³ module with geometry modifications was used. The main aim was to test the fire seal pipes on blast wall which was mounted using adaptor frame on the Gexcon module. The series of tests included some reference tests to fix the internal geometry layout/configuration that could give the desired explosion loading on the blast wall and pipe setup.

- **Location:** Sutra, Burgen
- **Year:** 11/04/2021 - 25/04/2021
- **Number of cases in series:** 10

**SELECTION OF TESTS**

- **Available measurements:** Pressure

**SIMULATION SETUP**

- **Grids used:** 167 mm

**REFERENCES**

[1] Brian A. Wilkins (2021), Explosion Testing of FireSeal Pipe Penetrations Seals and Blast Wall
Flacs-Cfd 22.2

Date: FEbRUAr 15, 2023

Postprocess quantities
Pressure

<table>
<thead>
<tr>
<th>Fuel</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propane</td>
<td>62</td>
<td>0.72</td>
<td>1.4</td>
<td>0.73</td>
<td>-0.43</td>
<td>0.5</td>
</tr>
<tr>
<td>Methane</td>
<td>7</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-0.0068</td>
<td>0.015</td>
</tr>
<tr>
<td>All fuels</td>
<td>69</td>
<td>0.75</td>
<td>1.4</td>
<td>0.75</td>
<td>-0.42</td>
<td>0.52</td>
</tr>
</tbody>
</table>

Geometric variance

Geometric mean bias

Simulation [barg]

Experiment [barg]
The experiments were conducted in an empty 64 m³ chamber with dimensions 4.6 m x 4.6 m x 3 m with square vents of 5.4 m² and 2.7 m². The fuels used were methane, propane and hydrogen. Location: Year: Number of cases in series: 18

Available measurements: Pressure

Grids used: 200 mm


## FLACS-CFD 22.2

**DATE: FEBRUAR 15, 2023**

**POSTPROCESS QUANTITIES**

### Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROPANE</td>
<td>6</td>
<td>1.1</td>
<td>14</td>
<td>0.17</td>
<td>0.39</td>
<td>1.4</td>
</tr>
<tr>
<td>HYDROGEN</td>
<td>6</td>
<td>2</td>
<td>2.6</td>
<td>0.33</td>
<td>0.87</td>
<td>1.8</td>
</tr>
<tr>
<td>METHANE</td>
<td>6</td>
<td>1.2</td>
<td>4.2</td>
<td>0.17</td>
<td>0.59</td>
<td>1.4</td>
</tr>
<tr>
<td>All fuels</td>
<td>18</td>
<td>1.4</td>
<td>5.4</td>
<td>0.22</td>
<td>0.68</td>
<td>1.9</td>
</tr>
</tbody>
</table>

![Graphs showing geometric variance and mean bias](image-url)
Vented explosion experiments were conducted using stoichiometric methane-air and propane-air mixtures in FM Global's large-scale explosion test chamber. The test chamber had overall dimensions of 4.6 x 4.6 x 3.0 m and an overall volume of 63.7 m³. A square vent with two possible surface areas, either 5.4 m² or 2.7 m², was located on one of the chamber's vertical walls. The tests were focused on the effect of fuel, enclosure size, ignition location, vent size, and obstacles on the pressure development of a propagating flame in a vented enclosure. The dependence of the maximum pressure generated on the experimental parameters was analysed.

**Location:** Rhode Island, USA  
**Year:** 2010  
**Number of cases in series:** 10

**REFERENCES**


2. **Bauwens, C.R. (2010),** Effect of ignition location, vent size, and obstacles on vented explosion overpressures in propane-air mixtures, *Combustion Science and Technology*
### POSTPROCESS QUANTITIES

<table>
<thead>
<tr>
<th>Fuel</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methane</td>
<td>6</td>
<td>1.7</td>
<td>2.3</td>
<td>0.17</td>
<td>0.76</td>
<td>1.6</td>
</tr>
<tr>
<td>Propane</td>
<td>4</td>
<td>1.3</td>
<td>2.3</td>
<td>0.5</td>
<td>0.54</td>
<td>1.1</td>
</tr>
<tr>
<td>All fuels</td>
<td>10</td>
<td>1.5</td>
<td>2.3</td>
<td>0.3</td>
<td>0.69</td>
<td>1.5</td>
</tr>
</tbody>
</table>

#### FLACS-CFD v22.2 User’s Manual

10.3 Validation cases
10.3.2.13  FMGlobal Hydrogen Initial Turbulence

**FMGLOBAL HYDROGEN INIT TURB**

**HYDROGEN**

The Experiment is conducted in 64 m³ chamber with dimensions 4.6 x 4.6 x 3 m with square vent of 5.4 m². Fuel used is Hydrogen with different concentration, initial turbulence and ignition location.

Location:  
Year:  
Number of cases in series: 15

**SELECTION OF TESTS**

Available measurements: Pressure

**SIMULATION SETUP**

Grids used: 200 mm

**REFERENCES**

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>15</td>
<td>8</td>
<td>87</td>
<td>0</td>
<td>1.5</td>
<td>7.2</td>
</tr>
</tbody>
</table>

### Pressure Validation Cases

#### Geometric Mean Bias vs. Geometric Variance

![Graph showing geometric mean bias vs. geometric variance](image)

#### Simulation vs. Experiment

![Graph showing comparison between simulation and experiment](image)
The experiments were conducted in a 64 m³ chamber with dimensions 4.6m x 4.6m x 3m with a square vent of 5.4 m² or 2.7 m². The fuel used was hydrogen with different concentrations and ignition locations [1].

**Location:**
- Year:
- Number of cases in series: 22

**Available measurements:** Pressure

**Grids used:** 200 mm

**REFERENCES**

10.3 Validation cases
## 10.3.2.15 Hydrogen Balloon

### HYDROGEN BALLOON

The experiment consisted of a 20 meters diameter polyethylene hemispheric balloon placed on ground and filled with stoichiometric hydrogen-air. The combustion was initiated by ignition pills at the centre of the hemisphere basement. Pressure dynamics were recorded using 11 transducers, installed on the ground level in a radial direction at distances from 2 to 80 meters away from the centre of the hemisphere basement at radii $R = 2.0, 3.5, 5.0, 6.5, 8.0, 10.0, 25.0, 35.0, 60.0$ and $80.0$ meters. Flame propagation was evaluated along the radial path between $45^\circ$ and $135^\circ$ from the point of ignition.

**Location:** Germany  
**Year:** 1983  
**Number of cases in series:** 1

### SELECTION OF TESTS

**Available measurements:** Pressure

### SIMULATION SETUP

**Grids used:** 600 mm

### REFERENCES

1. HySafe (2005), *SBEP-V2: Fh-ICT Balloon Test*
### Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>6</td>
<td>11</td>
<td>1</td>
<td>1</td>
<td>0.036</td>
<td>0.045</td>
</tr>
</tbody>
</table>

#### Geometric variance

![Geometric variance graph](image)

#### Simulation vs. Experiment

![Simulation vs. Experiment graph](image)

---

**10.3 Validation cases**

[FLACS-CFD v22.2 User’s Manual](#)
HYDROGEN REFUELLING STATION

HYDROGEN

The experiment in a mock-up of a hydrogen refuelling station was conducted jointly by Shell Global Solutions (UK) and the Health and Safety Laboratory (UK) in order to study the potential hazards and consequences associated with a hydrogen-air mixture explosion. The scenario of a stoichiometric hydrogen-air mixture explosion was offered for this simulation exercise.

Location: Shell Global Solutions (UK)
Year: 2007
Number of cases in series: 1

Layout of the experimental rig and mock-up car.

SELECTION OF TESTS

Available measurements: Pressure

SIMULATION SETUP

Grids used: 110 mm

REFERENCES

POSTPROCESS QUANTITIES

Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>9</td>
<td>0.81</td>
<td>1.1</td>
<td>1</td>
<td>-0.23</td>
<td>0.12</td>
</tr>
</tbody>
</table>

[Graph showing geometric mean bias and simulation vs. experiment for different fuels.]

10.3 Validation cases
66 vented hydrogen deflagration experiments were performed in 20-foot ISO containers. 34 tests were conducted with homogeneous and quiescent mixtures. The project objective was to improve the hydrogen safety for energy applications through pre-normative research on vented deflagrations.

Location: Gexcon AS (NORWAY)
Year: 2016-2018
Number of cases in series: 36

Available measurements: Pressure

Grids used: 100 mm, 153 mm

REFERENCES


### POSTPROCESS QUANTITIES

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>36</td>
<td>2.6 – 4.6</td>
<td>28 – 45</td>
<td>0.28 – 0.31</td>
<td>1.7</td>
<td>25 – 30</td>
</tr>
</tbody>
</table>

#### Diagrams

**Geometric mean bias vs. Geometric variance**

- **Simulation** vs **Experiment** [barg]

- **Red markers** indicate HYDROGEN.
To investigate vented hydrogen explosions in installations such as gas cabinets, cylinder enclosures, dispensers, and backup power systems. The experimental setup allows researchers to study the effect of internal obstacles (bottles), construction materials (including structural response), and the effect of mitigating hydrogen deflagrations by means of various venting devices. Experiments on small scale enclosure additionally provide information on the opening inertia of the vent panel at low hydrogen concentration and on the measurement of the structural response [1].

Location: University of Pisa (Italy)
Year: 2017
Number of cases in series: 9

Available measurements: Pressure

Grids used: 90 mm

REFERENCES
FLACS-CFD 22.2

DATE: FEBRUARY 15, 2023

POSTPROCESS QUANTITIES

Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>9</td>
<td>1.1</td>
<td>1</td>
<td>1</td>
<td>0.045</td>
<td>0.021</td>
</tr>
</tbody>
</table>

![Graphs and diagrams showing geometric mean bias and simulation vs experiment for HYDROGEN fuel.](image-url)
Gexcon AS developed a Joint Industry Project, Modelling Escalating Accident Scenarios and the Use of Risk-reducing technology for Explosion safety (MEASURE), that included the conduction of large scale experiments to provide validation data for explosion modelling (e.g., FLACS-CFD). Here, we consider the tests using a configuration with type 8 congestion (an array of 8x8x4 pipes of 168.3 mm in diameter and 655 mm in pitch, offering approximately 17% volume blockage) for eight different separation distances (SDs).

Location: United Kingdom
Year: 2013-2017
Number of cases in series: 2

### Selection of Tests

**Available measurements:** Flame arrival time, Pressure

### Simulation Setup

**Grids used:** 200 mm

### References

### Flame arrival time

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROPANE</td>
<td>70</td>
<td>0.73</td>
<td>1.2</td>
<td>0.97</td>
<td>-0.34</td>
<td>0.14</td>
</tr>
</tbody>
</table>

### Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROPANE</td>
<td>19</td>
<td>0.49</td>
<td>2.5</td>
<td>0.68</td>
<td>-1</td>
<td>9.4</td>
</tr>
</tbody>
</table>
METHANE, PROPANE, ETHYLENE

The project MERGE (Modelling and Experimental Research into Gas Explosions) was undertaken to address key issues relating to the influence on over-pressure and flame acceleration from varying the volume blockage, pipe diameters and fuels for particular obstacle configurations. The experiments of MERGE involved a single array of pipes contained in a homogeneous gas cloud, enclosed in a polythene tent. The obstacles used were referred as type A, B, C, D, E and CS.

Location: British Gas Fauld and Spadeadam test facilities (UK)
Year: 1991-1993
Number of cases in series: 19

Available measurements: Flame arrival time, Pressure

Grids used: 150 mm, 300 mm

British Gas (BG) (1992), Results from experiments with obstacle type A. British Gas Research & Technology Report.
British Gas (BG) (1992), Results from experiments with obstacle type B. British Gas Research & Technology Report.
British Gas (BG) (1991), Results from experiments with obstacle type C. British Gas Research & Technology Report.
British Gas (BG) (1992), Results from experiments with obstacle type D. British Gas Research & Technology Report.
British Gas (BG) (1993), Results from experiments with obstacle type E. British Gas Research & Technology Report.
British Gas (BG) (1993), Results from experiments with obstacle type CS. British Gas Research & Technology Report.
## Flame arrival time

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHANE</td>
<td>19</td>
<td>1 – 11</td>
<td></td>
<td>0.95 – 1</td>
<td>0.0038 – 0.028</td>
<td>0.0056 – 0.048</td>
</tr>
<tr>
<td>PROPANE</td>
<td>19</td>
<td>0.99 – 11</td>
<td>1 – 11</td>
<td>0.9 – 1</td>
<td>-0.1 – 0.043</td>
<td>0.014 – 0.057</td>
</tr>
<tr>
<td>METHANE, PROPANE</td>
<td>18</td>
<td>0.86 – 0.93</td>
<td>1 – 12</td>
<td>0.89 – 1</td>
<td>-0.26 – -0.1</td>
<td>0.04 – 0.13</td>
</tr>
<tr>
<td>All fuels</td>
<td>57</td>
<td>0.98 – 1</td>
<td>1 – 11</td>
<td>0.91 – 1</td>
<td>-0.12 – -0.013</td>
<td>0.021 – 0.081</td>
</tr>
</tbody>
</table>

## Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHANE</td>
<td>15</td>
<td>0.75 – 11</td>
<td>11</td>
<td>0.93 – 0.95</td>
<td>-0.99 – -0.075</td>
<td>0.044 – 0.32</td>
</tr>
<tr>
<td>PROPANE</td>
<td>19</td>
<td>0.45 – 0.57</td>
<td>21 – 29</td>
<td>0.38 – 0.53</td>
<td>-1.1 – -0.91</td>
<td>1.9 – 5</td>
</tr>
<tr>
<td>METHANE, PROPANE</td>
<td>20</td>
<td>0.56 – 2.5</td>
<td>1.6 – 3.9</td>
<td>0.5 – 0.76</td>
<td>-0.81 – 1</td>
<td>2.1 – 2.7</td>
</tr>
<tr>
<td>ETHYLENE</td>
<td>10</td>
<td>0.89</td>
<td>1</td>
<td>1</td>
<td>0.12</td>
<td>0.018</td>
</tr>
<tr>
<td>All fuels</td>
<td>54</td>
<td>0.59 – 12</td>
<td>1.5 – 2.5</td>
<td>0.63 – 0.72</td>
<td>-0.73 – -0.53</td>
<td>1.8 – 4.3</td>
</tr>
</tbody>
</table>

---

10.3 Validation cases
The campaign consists of series of gas explosion experiments conducted in a channel $1.5m \times 0.3m \times 0.3m$ to study the effect of vegetation in flame propagation. The effect of vegetation both in terms of tree species, number of branches and foliage presence is investigated. The propane-air mixture was filled in the channel with ignition source at centre of the closed end. Four pressure transducers were used with one fitted at closed end while other three placed along the rear wall. 

Location: 
Year: 
Number of cases in series: 6

Available measurements: Pressure

Grids used: 20 mm

REFERENCES

Hisken, Helene and Enstad, G.A. and Skjold, T. and Brewerton, R. (2013), The Effect of Vegetation with Various Degrees of Foliage on Gas Explosions in a $1.5m$ Channel
10.3 Validation cases
### MOGELEG CHANNEL

**HYDROGEN, N2**

A series of 56 tests were performed in a 0.13m³ test chamber at the Gexcon test laboratory at Fantoft, near Bergen, Norway. The vessel is rectangular in shape and is 1.44m long, 0.3m high and 0.3m wide. The test vessel contained baffle plate obstructions for turbulence generation. The vessel is closed on all sides and at one end. The fuel mixtures were usually ignited near the centre of the closed end of the chamber. All explosion venting occurred at the end of the vessel furthest from ignition. A limited number of tests were performed however, in which the ignition source was mounted at the upstream face of one of the baffle obstructions. The internal geometry of the vessel could be varied by the insertion of baffle plate obstructions of 5cm, which were fixed to the walls and floor of the vessel. Due to the high reactivity of the gas mixtures to be used, and to maintain integrity of the test chamber, 2 obstacle configurations (2 baffles and 4 baffles) were used in the current experiments in addition to tests performed in the empty chamber without obstructions. 4 gases were used in the tests (H₂, 3H₂+CO, H₂+CO, 3H₂+N₂) at different concentrations in air.

**Location:** Gexcon AS, Bergen, Norway  
**Year:** 2001-2002  
**Number of cases in series:** 33

### SELECTION OF TESTS

**Available measurements:** Pressure

### SIMULATION SETUP

**Grids used:** 40 mm

### REFERENCES

### FLACS-CFD 22.2

**DATE: FEBRUAR 15, 2023**

**POSTPROCESS QUANTITIES**

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>108</td>
<td>0.74</td>
<td>2</td>
<td>0.68</td>
<td>0.13</td>
<td>31</td>
</tr>
<tr>
<td>HYDROGEN, N2</td>
<td>24</td>
<td>0.83</td>
<td>1.1</td>
<td>0.92</td>
<td>-0.11</td>
<td>0.072</td>
</tr>
<tr>
<td>All fuels</td>
<td>132</td>
<td>0.75</td>
<td>1.8</td>
<td>0.72</td>
<td>0.086</td>
<td>2.5</td>
</tr>
</tbody>
</table>

|       | 10^{-1} | 10^{0} | 10^{1} | | 5   | 10  | 15  |
|-------|---------|--------|--------||------|-----|-----|
| Geometric mean bias|         |        |        |
| Geometric variance  |         |        |        |
| Simulation [barg]   |         |        |        |
| Experiment [barg]   |         |        |        |

**Graphs:**

- Geometric variance vs. Geometric mean bias
- Simulation vs. Experiment [barg]
The campaign consists of thirteen gas explosion tests conducted in a cylindrical vessel of diameter 2m and length 3.5m. The vessel had a rectangular hole 1.7m high and 0.8m wide covered with a sturdy paper at one of the edges. Propane-air mixture was used as fuel which was ignited by a point source located at 0.4m from the covered edge on the vessel axis. Obstacles were arranged in arrays in the diametral plane of the vessel. The internal pressure was measured by two pressure transducers.

Location:
Year:
Number of cases in series: 14

Available measurements: Pressure

Grids used: 200 mm

V. V. Mol'kov and V. V. Agafonov and S. V. Aleksandrov (1997), Deflagration in a vented vessel with internal obstacles, *Combustion, Explosion and Shock Waves*
### POSTPROCESS QUANTITIES

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROPANE</td>
<td>14</td>
<td>1.7</td>
<td>1.4</td>
<td>0.86</td>
<td>0.56</td>
<td>0.44</td>
</tr>
</tbody>
</table>

**Pressure**

![Graphs showing geometric variance and simulation vs. experiment data for PROPANE fuel.](image)

**10.3 Validation cases**

**FLACS-CFD v22.2 User’s Manual**
METHANE
Campaign consists of series of explosion test conducted inside rectangular chambers with 700mm x 700mm cross-section and a large top-venting area of 700mm x 210mm. The height of chamber was varied from 200mm to 1000mm. Three different multiple obstacles with square, circular and triangular cross-sections were used with diameters/size 70mm and 100mm respectively. Methane-air mixture in all tests was ignited near the bottom wall. Pressure values were recorded with transducer mounted on top wall of chamber, 20mm from exit.

Location:
Year:
Number of cases in series: 20

PARK

Available measurements: Pressure

SIMULATION SETUP
Grids used: 13 mm, 25 mm, 33 mm

REFERENCES
Park, Dal and Lee, Ys and Green, Anthony (2008), Experiments on the effects of multiple obstacles in vented explosion chambers, Journal of hazardous materials
### Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHANE</td>
<td>4</td>
<td>0.31 – 2.1</td>
<td>11</td>
<td>0 – 1</td>
<td>-1.1 – 0.77</td>
<td>0.11 – 1.7</td>
</tr>
</tbody>
</table>

#### Validation cases 545
A series of vented explosion tests has been carried out using a 30m³ explosion chamber. One end of the chamber was provided with a square orifice, a series of area reducing plates with vent area 1/2, 1/4, and 1/8 of the area of the end face were used. Fuel used were propane and natural gas ignited within the chamber at different ignition locations, i.e., center, front, and end of the chamber[^3].

**Location:**

- **Year:**
- **Number of cases in series:** 16

**Selection of Tests**

- Available measurements: Pressure

**Simulation Setup**

- Grids used: 140 mm

**References**

[^3]: A. J. Harrison and J. A. Eyre (1987), External Explosions† as a Result of Explosion Venting, Combustion Science and Technology
## FLACS-CFD 22.2

**DATE:** FEBRUAR 15, 2023

### POSTPROCESS QUANTITIES

#### Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETHANE, METHANE</td>
<td>12</td>
<td>0.71</td>
<td>17</td>
<td>0.86</td>
<td>-0.48</td>
<td>0.84</td>
</tr>
<tr>
<td>PROPANE</td>
<td>13</td>
<td>0.59</td>
<td>15</td>
<td>0.69</td>
<td>-0.47</td>
<td>0.53</td>
</tr>
<tr>
<td>All fuels</td>
<td>25</td>
<td>0.64</td>
<td>16</td>
<td>0.64</td>
<td>-0.47</td>
<td>0.65</td>
</tr>
</tbody>
</table>

![Graphs showing geometric variance and geometric mean bias against experiment values.](image.png)
Hydrogen-air mixtures were ignited in a 3 m x 3 m x 2 m congested area at the Health and Safety Laboratory. Three different levels of congestion were used by varying the amount of repeated pipes in the rig. Both rich and lean hydrogen mixtures were used.

**Location:** UK, Health and Safety Laboratory  
**Year:** 2006  
**Number of cases in series:** 7

**Available measurements:** Flame arrival time, Pressure

**Grids used:** 133 mm

**REFERENCES**

**Flame arrival time**

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>45</td>
<td>1.8</td>
<td>1.7</td>
<td>0.64</td>
<td>0.52</td>
<td>0.36</td>
</tr>
</tbody>
</table>

**Pressure**

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>47</td>
<td>1.3</td>
<td>1.2</td>
<td>0.96</td>
<td>0.23</td>
<td>0.16</td>
</tr>
</tbody>
</table>
Hydrogen-air mixtures were ignited in a 3 m x 3 m x 2 m congested area at the Health and Safety Laboratory. The congested area consisted of repeated pipes. The amount of hydrogen in the experiments was varied from 0 (100% methane) to 100 (0% methane).

**Location:** UK, Health and Safety Laboratory  
**Year:** 2006  
**Number of cases in series:** 5

**Available measurements:** Pressure

**Grids used:** 133 mm

**References**
<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>1</td>
<td>2.5</td>
<td>2.3</td>
<td>0</td>
<td>0.86</td>
<td>0.91</td>
</tr>
<tr>
<td>METHANE</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.021</td>
<td>0.00044</td>
</tr>
<tr>
<td>HYDROGEN, METHANE</td>
<td>3</td>
<td>2.6</td>
<td>2.5</td>
<td>0</td>
<td>0.85</td>
<td>1</td>
</tr>
<tr>
<td>All fuels</td>
<td>5</td>
<td>2.1</td>
<td>2</td>
<td>0.2</td>
<td>0.83</td>
<td>1.6</td>
</tr>
</tbody>
</table>

**Graphs:**
- Geometric variance vs. geometric mean bias
- Simulation vs. experiment [barg]

**Colors and Symbols:**
- Red: HYDROGEN, METHANE
- Blue: METHANE
- Orange: HYDROGEN
- Green: Detonation
- Black: Other
SOLVEX

METHANE, PROPANE

The campaign consisted of gas explosions in large scale enclosure with size 10m*8.75m*6.25m and vent located at centre of the front wall. The experiments were performed with propane-air and methane-air mixtures inside the enclosure ignited by a point ignition source located at centre of the rear wall. The tests were conducted with empty enclosure configuration, one or two rows of 0.5m diameter pipes.[1]

Location: Shell UK
Year: 
Number of cases in series: 8

SELECTION OF TESTS

Available measurements: Pressure

SIMULATION SETUP

Grids used: 386 mm

REFERENCES


## Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHANE, PROPANE</td>
<td>4</td>
<td>21</td>
<td>1.9</td>
<td>0.5</td>
<td>0.83</td>
<td>1.4</td>
</tr>
<tr>
<td>PROPANE</td>
<td>4</td>
<td>1.9</td>
<td>1.5</td>
<td>0.75</td>
<td>0.68</td>
<td>0.83</td>
</tr>
<tr>
<td>All fuels</td>
<td>8</td>
<td>2</td>
<td>1.7</td>
<td>0.62</td>
<td>0.74</td>
<td>1.1</td>
</tr>
</tbody>
</table>

![Graph showing geometric variance vs geometric mean bias](image1)

![Graph showing simulation vs experiment](image2)
SRI-CONFINED-TUBE

HYDROGEN

The facility consists of a square section steel tube with an inside dimension of 38.1 cm and a length of 990 cm. The ignition end of the tube was closed with a steel plate and the opposite end was opened prior to ignition by piercing a latex rubber diaphragm stretched tightly over the end. Various obstacle configurations were used to promote turbulence of the flame propagation. Figure 1 is a cross section of the tube showing the layout of the obstacles. The steel obstacles were 6.35 cm thick and were attached to a steel floor that was 1.27 cm thick on top of 1.91-cm-thick risers. Blockage ratios of 0.32, 0.47, and 0.65 were obtained using blocks that had heights of 11.43 cm, 16.51 cm, and 22.86 cm. The blocks were spaced at regular intervals of 38.1 cm, 76.2 cm, or 152.4 cm for tests using 25 blocks, 13 blocks, or 7 blocks respectively. The first block was always 38.1 cm from the initiation end.

Location: DoE, USA
Year: 2002
Number of cases in series: 8

SELECTION OF TESTS

Available measurements: Flame arrival time, Pressure

SIMULATION SETUP

Grids used: 20 mm

REFERENCES

554 Validation
10.3.2.29 SRI Confined Tube

FLACS-CFD v22.2 User’s Manual
10.3.2.30  TNO 39m3

METHANE
The experiments were conducted in a concrete enclosure of size 4mx3.7mx2.6m with an internal volume of 38.5m³. Methane-air mixture was filled in the enclosure through three locations. Ignition source varied across the campaign with locations at rear, centre and near vent. Three pressure transducers were mounted inside the enclosure. Different vent configurations were used with single or multiple layers of polyethylene sheet and single plaster sheet plate as vent cover. Few tests were conducted with a single large obstacle placed at centre of the enclosure.[]

Location: Year: Number of cases in series: 6

Available measurements: Pressure

Grids used: 173 mm

Outlet of Gaseous Explosions

W.P.M. Mercx, C.J.M. van Wingerden, H.J. Pasman (1992), Venting of Gaseous Explosions
### Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>Vg</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHANE</td>
<td>8</td>
<td>0.093</td>
<td>4.1e+02</td>
<td>0</td>
<td>-1.7</td>
<td>11</td>
</tr>
</tbody>
</table>

#### Geometric mean bias vs. Simulation [ barg ]

10.3 Validation cases

---

**FLACS-CFD v22.2 User’s Manual**
### TRAFFIC TUNNEL SQUARE

**HYDROGEN**

Hydrogen explosion in a traffic tunnel (with and without cars). The experimental set-up consisted of a 78.5m long tunnel with a diameter of 2.4 m and a cross-sectional area of 3.74 m². The experimental facility was a one-fifth scale mock-up of a typical tunnel for road transport. The explosive hydrogen-air mixture was located in a 10 m long region, filling a volume equal to 37 m³. Several experiments were carried out, using different hydrogen concentration. The experiments with 30% hydrogen volumetric concentration in air have been considered. The geometry set-up was either an empty tunnel or a tunnel with 4 vehicle models on the floor centreline inside the hydrogen-air mixture. The vehicle models measured 0.94 m in length, 0.362 m in width and 0.343 m in height, representing typical real-vehicles at one-fifth scale. The distance between vehicles was 1.52 m. The blockage ratio due to the presence of the vehicles was 0.03. Pressure transducers were mounted on the side wall of the tunnel along its entire length. The ignition position was located in the middle of the tunnel for all experiments.

**Location:**

**Year:** 2009

**Number of cases in series:** 2

#### SELECTION OF TESTS

**Available measurements:** Pressure

#### SIMULATION SETUP

**Grids used:** 133 mm

#### REFERENCES

### FLACS-CFD 22.2

**DATE:** FEBRUARY 15, 2023

**POSTPROCESS QUANTITIES**

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>3</td>
<td>0.81</td>
<td>1</td>
<td>1</td>
<td>-0.21</td>
<td>0.047</td>
</tr>
</tbody>
</table>

**10.3 Validation cases**

---

**FLACS-CFD v22.2 User's Manual**

---

[Diagram of geometric mean bias and simulation vs. experiment]
Table 10.2: Dispersion validation cases.

<table>
<thead>
<tr>
<th>Case name</th>
<th>Type</th>
<th>Variable</th>
<th>N</th>
<th>SMG</th>
<th>VG</th>
<th>Mexic</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Barro</td>
<td>Dispersion</td>
<td>Fuel volume fraction</td>
<td>10</td>
<td>0.24</td>
<td>1.3e+05</td>
<td>0.04</td>
<td>0.70</td>
<td>2.4</td>
<td></td>
</tr>
<tr>
<td>CABIC</td>
<td>Dispersion</td>
<td>Fuel volume fraction</td>
<td>10</td>
<td>0.4</td>
<td>1.8</td>
<td>0.15</td>
<td>4</td>
<td>4.1</td>
<td></td>
</tr>
<tr>
<td>Coyote</td>
<td>Dispersion</td>
<td>Fuel volume fraction</td>
<td>15</td>
<td>0.001</td>
<td>1.0e+10</td>
<td>0.25</td>
<td>0.14</td>
<td>0.18</td>
<td></td>
</tr>
<tr>
<td>RESEA HOMOGENEOUS RELEASE</td>
<td>Dispersion</td>
<td>Fuel volume fraction</td>
<td>40</td>
<td>1.1–1.5</td>
<td>1.1–1.2</td>
<td>0.94–0.96</td>
<td>0.075–0.38</td>
<td>0.622–0.13</td>
<td></td>
</tr>
<tr>
<td>KriGox</td>
<td>Dispersion</td>
<td>Fuel volume fraction</td>
<td>155</td>
<td>1.1</td>
<td>1.0</td>
<td>0.73</td>
<td>0.83</td>
<td>0.62</td>
<td></td>
</tr>
<tr>
<td>Mogul landslide</td>
<td>Dispersion</td>
<td>Fuel volume fraction</td>
<td>10</td>
<td>0.98</td>
<td>1.1</td>
<td>0.5</td>
<td>0.14</td>
<td>0.639</td>
<td></td>
</tr>
<tr>
<td>Warehouse sandia release</td>
<td>Dispersion</td>
<td>Fuel volume fraction</td>
<td>6</td>
<td>1.1</td>
<td>1.1</td>
<td>0.81</td>
<td>0.14</td>
<td>0.637</td>
<td></td>
</tr>
<tr>
<td>Thames island</td>
<td>Dispersion</td>
<td>Fuel volume fraction</td>
<td>15</td>
<td>0.08</td>
<td>1.0</td>
<td>0.87</td>
<td>0.032</td>
<td>0.21</td>
<td></td>
</tr>
<tr>
<td>LGMI Falcon</td>
<td>Dispersion</td>
<td>Fuel volume fraction</td>
<td>3</td>
<td>0.52</td>
<td>2.0</td>
<td>0.93</td>
<td>1.2</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

10.3.3 Dispersion

Figure 10.3 (top-centre) summarises the presented dispersion validation cases in a single scatter plot, while table 10.2 gives a summary of the individual experiments. Performance is very good across the different fuel types and ambient conditions. In general, the outliers can be explained by challenges modelling the variation of wind speed and direction in large-scale dispersion experiments. Measurements in experiments are typically across an arc-wise array of sensors. Narrow plumes sometimes miss all of the sensors on an arc, the resulting low concentration at the corresponding arc-distance causes problems in calculating the geometric mean and variance.
### 10.3 Validation cases

#### 10.3.3.1 Burro Tests

<table>
<thead>
<tr>
<th>BURRO</th>
<th>METHANE</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Burro field experiments consisted of nine large spills of LNG onto a 1-m-deep pool of water. The four selected test cases are those which have been most extensively analysed and cover the widest range of meteorological and spill conditions.</td>
<td></td>
</tr>
</tbody>
</table>

**Location:** China Lake testing site, California, U.S.  
**Year:** 1980  
**Number of cases in series:** 4

<table>
<thead>
<tr>
<th>SELECTION OF TESTS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Available measurements:</strong> Fuel volume fraction</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SIMULATION SETUP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Grids used:</strong> 625 mm</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>REFERENCES</th>
</tr>
</thead>
</table>
FLACS-CFD 22.2
DATE: FEBRUAR 15, 2023
POSTPROCESS QUANTITIES
Fuel volume fraction

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHANE</td>
<td>14</td>
<td>0.34</td>
<td>7.1e+02</td>
<td>0.64</td>
<td>-0.39</td>
<td>1.4</td>
</tr>
</tbody>
</table>

FLACS-CFD v22.2 User’s Manual
CHRC carried out wind-tunnel experiments on the dispersion of carbon dioxide (CO2), both with and without obstacles. One of the aims of the experiments was to produce validation data for the FEM3A dispersion model. The wind tunnel was an ultra-low-speed boundary-layer wind tunnel capable of simulating the constant stress layer of the atmospheric boundary layer. Airflow from the driving fans passed through a circular-to-rectangular transition to a 7 ft. high, 20 ft. wide and 80 ft. long working area in which the floor was covered with smooth rubber matting on which roughness elements were mounted. The obstacles, a dike was square with an inner dimension of 63 cm and a wall height of 3.7 cm and the model tank was 31 cm in diameter with a hemi-spherical dome top and an overall height of 28.3 cm.

**Location:** University of Arkansas, USA,  
**Year:** 1988  
**Number of cases in series:** 3

### Selection of Tests

**Available measurements:** Fuel volume fraction

### Simulation Setup

**Grids used:** 6 mm

### References

**FLACS-CFD 22.2**

**DATE: FEBRUAR 15, 2023**

**POSTPROCESS QUANTITIES**

**Fuel volume fraction**

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO2, PROPANE</td>
<td>16</td>
<td>0.4</td>
<td>2.6</td>
<td>0.25</td>
<td>-1</td>
<td>4.3</td>
</tr>
</tbody>
</table>

**Validation**

**FLACS-CFD v22.2 User's Manual**
10.3 Validation cases

10.3.3.3 Coyote Tests

**COYOTE**

**METHANE**

The Coyote field experiments consisted of ten large spills of LNG onto a 1-m-deep pool of water. The three selected test cases are those which have been most extensively analyzed and are regarded as benchmarks for dispersion model validation.

**Location:** China Lake testing site, California, U.S.
**Year:** 1981
**Number of cases in series:** 3

**SELECTION OF TESTS**

Available measurements: Fuel volume fraction

**SIMULATION SETUP**

Grids used: 40 mm

**REFERENCES**


**FLACS-CFD 22.2**

**DATE: FEBRUARY 15, 2023**

**POSTPROCESS QUANTITIES**

**Fuel volume fraction**

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHANE</td>
<td>15</td>
<td>0.093</td>
<td>2.6e+19</td>
<td>0.73</td>
<td>-0.14</td>
<td>0.19</td>
</tr>
</tbody>
</table>

![Graph showing geometric mean bias and experiment data for METHANE fuel.](image)
## 10.3.3.4 HySEA Inhomogeneous Release

**HYSEA INHOMOGENEOUS RELEASE**

**HYDROGEN**

66 vented hydrogen deflagration experiments were performed in a 20-foot ISO containers. 22 tests were performed involving either an empty container with only the frame inserted, or a pipe rack installed in centre position inside the container. The experiments involved release of hydrogen inside the container either from a circular pipe or from a cubical box located at floor centre of the container and above floor and later ignited with ignition location on the upper back wall.

<table>
<thead>
<tr>
<th>Location: Gexcon AS (Norway)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year: 2016-2018</td>
</tr>
<tr>
<td>Number of cases in series: 12</td>
</tr>
</tbody>
</table>

**SELECTION OF TESTS**

Available measurements: Fuel volume fraction

**SIMULATION SETUP**

Grids used: 100 mm, 150 mm

**REFERENCES**


## Fuel volume fraction

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>48</td>
<td>1.1 – 1.5</td>
<td>1 – 1.2</td>
<td>0.94 – 0.98</td>
<td>0.075 – 0.38</td>
<td>0.022 – 0.15</td>
</tr>
</tbody>
</table>

![Graph showing geometric mean bias and simulation vs experiment](image-url)
10.3.3.5 LLNL Falcon Tests

The Falcon tests were performed to verify the effectiveness of vapour barriers in reducing the risk associated with the release and dispersion of LNG. The spill was distributed over the water pond by a 4-arms piping system. The spill pond was surrounded by a high fiberglass fence and upwind of the spill pond a billboard structure was mounted to reproduce the typical turbulence generated by a storage tank. The three Falcon test cases are those which are regarded as benchmarks.

Location: Franchman Flat (Nevada), U.S.
Year: 1987
Number of cases in series: 3

Available measurements: Fuel volume fraction

Grids used: 1000 mm

REFERENCES


**FLACS-CFD 22.2**

**DATE:** FEBRUARY 15, 2023

**POSTPROCESS QUANTITIES**

**Fuel volume fraction**

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHANE</td>
<td>3</td>
<td>0.32</td>
<td>5</td>
<td>0.33</td>
<td>-1.2</td>
<td>3</td>
</tr>
</tbody>
</table>

![Graphs showing geometric variance and simulated vs. experimental values](image_url)
Dense gas carbon dioxide (CO₂) was released at ground level for longer time periods (continuous “plumes”) and for shorter time (short-duration transient “puffs”), including both neutral and stable conditions. The desert surface, a dry lake bed known as Frenchman Flat, was artificially roughened using combinations of flat billboard obstacles in order to simulate the roughness of an industrial site and its surroundings at about 1/10 scale. The roughness elements were constructed from a plywood billboards and were categorized as Equivalent Roughness Pattern (ERP, 2.4m x 2.4m) and Uniform Roughness Array (URA, 0.2m high x 0.8m wide). The roughness elements were installed along the cross-wind (y direction) and along-wind (x direction). CO₂ was released from a 1.5m x 1.5m area source, with a nearly constant emission rate. 84 concentration monitors were installed on the four downwind arcs (25, 50, 100, and 225m). Meteorological instruments were installed on five towers with heights 24m (EPA), 4.9m (Met1), 8m (Met2), 4.9m (Met3), and 8m (Met4). Puff releases experiments comprised of 34, finite duration (20 seconds) tests. 13 tests were performed with URA and ERP in place, while 21 test with ERP removed.

**Location:** Frenchman Flat, Nevada, U.S.A.
**Year:** 1995
**Number of cases in series:** 51

### References

Hanna, S.R & Chang, J.C (2001), Use of the Kit Fox field data to analyze dense gas dispersion modeling issues, *Atmospheric Environment*
**FLACS-CFD 22.2**

**DATE: FEBRUAR 15, 2023**

**POSTPROCESS QUANTITIES**

**Fuel volume fraction**

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO2</td>
<td>153</td>
<td>1.1</td>
<td>1.6</td>
<td>0.73</td>
<td>0.03</td>
<td>0.62</td>
</tr>
</tbody>
</table>

![Graphs showing geometric variance and simulation vs. experiment data for CO2 fuel volume fraction.](image-url)
The Maplin Sands trials were conducted by Shell Research Limited in 1980 and consisted of 34 spills of liquefied gases onto the sea. Both continuous and instantaneous releases of LNG and LPG were carried out through a vertical pipe terminating above the water surface. The release site was an area of tidal sands in the Thames estuary so that the cloud dispersion occurred over flat terrain. The selected Maplin Sands test cases are three of the four releases in the Modelers Data Archive. Case 29 is omitted, since Ermak et al. stated that for this case sub-surface vaporization was considerable, leading to gas jetting as high as 10 m in the source area, such that specification of a vapor source term could prove problematic.

Location: Maplin Sands (Thames estuary), U.K.
Year: 1980
Number of cases in series: 3

Available measurements: Fuel volume fraction

Grids used: 50 mm

Mauri, L. (2016), Evaluation of FLACS Performance Against the Model Validation Database, appendix 1. Report number GexCon-14-48653-C-1
### Fuel volume fraction

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHANE</td>
<td>10</td>
<td>0.98</td>
<td>1.1</td>
<td>0.9</td>
<td>-0.14</td>
<td>0.059</td>
</tr>
</tbody>
</table>

#### Diagrams

- Geometric mean bias
- Simulation [%] vs Experiment [%]

---

574 Validation

FLACS-CFD v22.2 User’s Manual
In the Thorney Island continuous release experiments, a mixture of 32% R-12 and 68% nitrogen was released through a duct emerging from the ground and capped by a 2 m diameter plate. The resulting outflow had zero vertical momentum and low radial momentum flow, comparable to a gravity current velocity. Among the three trials, test number 46 was excluded because of the limited data due to changes in the wind direction during the experiment.

**Location:** Thorney Island, U.K.  
**Year:** June 15, 1984  
**Number of cases in series:** 2

### Available measurements:
Fuel volume fraction

### SIMULATION SETUP

| Grids used | 380 mm |

### REFERENCES

4. McQuaid, J. & Roe buck, B. (1985), Large-Scale Field Trials on Dense Vapour Dispersion.
### Fuel volume fraction

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>Vg</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BUTANE, PROPANE</td>
<td>15</td>
<td>0.98</td>
<td>1.6</td>
<td>0.67</td>
<td>-0.032</td>
<td>0.21</td>
</tr>
</tbody>
</table>

**Graphs**

- **Geometric variance**
  - Y-axis: Geometric variance
  - X-axis: Geometric mean bias

- **Simulation vs Experiment**
  - Y-axis: Simulation [vol%]
  - X-axis: Experiment [vol%]
  - Points: BUTANE, PROPANE
10.3 Validation cases

10.3.3.9 Warehouse Sandia Release

WAREHOUSE SANDIA RELEASE

HYDROGEN

The experiments were performed in a scaled model of a warehouse facility. The dimensions were 3.64 m wide by 4.99 m long by 2.72 m high, with a total volume of 45.4 m³. The mass of hydrogen was released into warehouse and the concentration variation with time was measured [1].

Location: Livermore, California
Year: 2011
Number of cases in series: 1

SELECTION OF TESTS

Available measurements: Fuel volume fraction

SIMULATION SETUP

Grids used: 26 mm

REFERENCES


### Fuel volume fraction

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>6</td>
<td>1.1</td>
<td>1.1</td>
<td>0.83</td>
<td>0.14</td>
<td>0.057</td>
</tr>
</tbody>
</table>

#### Diagrams

- **Geometric Variance vs. Geometric Mean Bias**
- **Simulation vs. Experiment [vol%]**

---

**FLACS-CFD v22.2 User’s Manual**

GEXCON
10.3 Validation cases

Table 10.3: Fire validation cases.

<table>
<thead>
<tr>
<th>Case name</th>
<th>Type</th>
<th>Variable</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>Monics</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gasoline/Sprinkler pool Fire</td>
<td>Fire</td>
<td>Radiative Flux</td>
<td>1.4</td>
<td>1.2</td>
<td>1</td>
<td>0.37</td>
<td>0.29</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GL Hydrogen Jet Fire</td>
<td>Fire</td>
<td>Flame length</td>
<td>2</td>
<td>0.96</td>
<td>1</td>
<td>-0.17</td>
<td>0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ISL hydrogen impinging Fire</td>
<td>Fire</td>
<td>Radiative Flux</td>
<td>26</td>
<td>0.62</td>
<td>0.62</td>
<td>0.09</td>
<td>1.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LNG Fire</td>
<td>Fire</td>
<td>Radiative Flux</td>
<td>22</td>
<td>2.2</td>
<td>4.8</td>
<td>0.41</td>
<td>0.84</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LNG/France</td>
<td>Fire</td>
<td>Radiative Flux</td>
<td>31</td>
<td>3.7</td>
<td>7.3</td>
<td>0.59</td>
<td>1.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Monitor Pool Fire</td>
<td>Fire</td>
<td>Radiative Flux</td>
<td>4</td>
<td>0.79</td>
<td>1.1</td>
<td>0.20</td>
<td>0.20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nonoxy Jet Fire</td>
<td>Fire</td>
<td>Flame trajectory</td>
<td>0.32</td>
<td>4.8</td>
<td>0.33</td>
<td>1.2</td>
<td>4.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Natural Jet Fire</td>
<td>Fire</td>
<td>Flame length</td>
<td>0.92</td>
<td>1</td>
<td>1</td>
<td>-0.10</td>
<td>-0.04</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sandia-Cryogen Hydrogen Jet Fires</td>
<td>Fire</td>
<td>Radiative Flux</td>
<td>24</td>
<td>0.81</td>
<td>1.1</td>
<td>0.08</td>
<td>0.27</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINTF Impinging Jet</td>
<td>Fire</td>
<td>Radiative Flux</td>
<td>25</td>
<td>1.7</td>
<td>1.5</td>
<td>0.88</td>
<td>0.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SINTF Impinging Jet</td>
<td>Fire</td>
<td>Radiative Flux</td>
<td>4</td>
<td>0.54</td>
<td>1.8</td>
<td>0.67</td>
<td>0.30</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10.3.4 Fire

Figure 10.3 (top-right) summarises the presented fire validation cases in a single scatter plot, while table 10.3 gives a summary of the individual experiments.

The FLACS-CFD fire solver generally performs well for jet fires including horizontal and vertical jet fires, with or without crosswind, open, or impinging. Generally, parameters like flame length, flame trajectory, flame temperature, radiative and total heat flux compare well against experiments. Fire simulations for large horizontal non-impinging jet fires are shown to be dominated by significantly more buoyancy forces on the end part of the flame, causing the flame to bend off and rise up earlier compared to the experimental flame. The FLACS-CFD fire solver performs well for both steady flow rate and blowdown scenarios for various species including hydrogen.
SANDIA CRYOGENIC HYDROGEN JET FIRES

The experiments were conducted at the Turbulent Combustion Laboratory of SNL in USA. The main aim of the experiments was to investigate the ignition and flame characteristics of cryogenic underexpanded jet fires. The analysed scenarios were concerned with hydrogen releases with temperature in the range 37-295 K and pressure 2-6 bar abs. The release temperature and pressure were maintained constant during each test and monitored upstream the interchangeable orifice of diameter 0.75 mm or 1 mm or 1.25 mm. The hydrogen was released vertically upward in the laboratory equipped with an exhaust gas collection system. The incident thermal radiation was monitored at 5 sensors located along the jet flame and at 0.2 m from the jet axis. Five tests with orifice diameter 1.25mm out of the entire set of experiments performed by SNL have been selected for the validation.

Location: Sandia National Laboratories, USA
Year: 2015
Number of cases in series: 5

SELECTION OF TESTS
Available measurements: Flame length, Radiative Flux

SIMULATION SETUP
Grids used: 100 mm

REFERENCES
### Flame length

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>5</td>
<td>1.1</td>
<td>1</td>
<td>1</td>
<td>0.11</td>
<td>0.013</td>
</tr>
</tbody>
</table>

### Radiative Flux

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>25</td>
<td>1.7</td>
<td>1.5</td>
<td>0.68</td>
<td>0.6</td>
<td>0.59</td>
</tr>
</tbody>
</table>

---

**FLACS-CFD v22.2 User’s Manual**
Two large-scale hydrogen jet fire experiments were conducted at the GL Noble Denton Spadeadam Test Site in North Cumbria, UK. Compressed hydrogen gas was released from a nominal 60 bar stagnation pressure through a horizontally orientated 1 m long stretch of pipe with respective internal diameters of 20.9 and 52.5 mm and located 3.25 m above the ground. Wind speed for test with smaller diameter nozzle was measured as 2.84 m/s at 68.5° from true north while for the other case wind speed was 0.83 m/s at 34° from true north. Incident thermal radiation was measured at 13 locations by wide-angle Medtherm radiometers mounted on tripods around the test area.

Location: GL Noble Denton Spadeadam Test Site, North Cumbria, UK
Year: 2008
Number of cases in series: 2

Available measurements: Flame length, Flame trajectory, Radiative Flux

Grids used: 4000 mm

### Postprocess Quantities

#### Flame length

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>2</td>
<td>0.86</td>
<td>1</td>
<td>1</td>
<td>-0.17</td>
<td>0.041</td>
</tr>
</tbody>
</table>

#### Radiative Flux

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>26</td>
<td>0.62</td>
<td>1.7</td>
<td>0.62</td>
<td>-0.39</td>
<td>1.2</td>
</tr>
</tbody>
</table>
A series of jet fire tests was performed to investigate the effectiveness of barrier walls at preventing radiation and physical transport of fire from hydrogen jet flames. Comparison of the reduction of jet-fire hazard by using 90deg and 60deg inclined walls to a free jet was done in terms of both thermal radiation and overpressures. Hydrogen was released at 200 barg through a nozzle of either 9.5, 6.4 or 3.2 mm.

Location: Health and Safety Laboratory (UK)
Year: 2009
Number of cases in series: 6

Available measurements: Radiative Flux

Grids used: 3 mm

FLACS-CFD 22.2

DATE: FEBRUAR 16, 2023

POSTPROCESS QUANTITIES

Radiative Flux

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>HYDROGEN</td>
<td>22</td>
<td>2.2</td>
<td>4.8</td>
<td>0.41</td>
<td>0.84</td>
<td>2.3</td>
</tr>
</tbody>
</table>

[Graphs showing geometric variance and geometric mean bias vs. experiment and simulation values for different fuels]
Large-scale turbulent natural gas/air diffusion flames were used to evaluate analysis of flame structure and radiation properties. Radiation from turbulent diffusion flames dominates energy transport from unwanted fires, influencing their burning and growth rates. The experiments involved vertically upward injection of natural gas (96% methane by volume) in still air. Seven flames were tested (flame heights approx. 25 m) with chemical energy release rates in the range of 135-210 MW.

**Location:** USA  
**Year:** 1986  
**Number of cases in series:** 7

**Selection of Tests**

**Available measurements:** Radiative Flux

**Simulation Setup**

**Grids used:** 250 mm

**References**

## POSTPROCESS QUANTITIES

### Radiative Flux

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHANE</td>
<td>31</td>
<td>1.7</td>
<td>2.4</td>
<td>0.39</td>
<td>0.7</td>
<td>11</td>
</tr>
</tbody>
</table>

![Graphs showing geometric variance and simulation vs. experiment data for METHANE.](image-url)
A series of six large scale high pressure, jet fires experiments were conducted using natural gas and natural gas-hydrogen (approximately 24% by volume) mixtures. For each fuel, the three tests involved horizontal releases from 20, 38 and 50mm diameter holes at a gauge pressure of approximately 60 bar. The fires also engulfed a 1m diameter horizontal pipe placed across the flow direction and about halfway along the flame. This pipe was instrumented to measure the heat fluxes to the pipe.

**Location:** GL Noble Denton Spadeadam Test Site, North Cumbria, UK

**Year:** 2005

**Number of cases in series:** 6

**Available measurements:** Flame length, Flame trajectory, Radiative Flux

**Grids used:** 100 mm, 180 mm, 190 mm, 260 mm

---

**References**

**Flame length**

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETHANE, HYDROGEN, METHANE, N2</td>
<td>1</td>
<td>0.87 – 0.99</td>
<td>1</td>
<td>1</td>
<td>-0.14 – -0.0092</td>
<td>8.5e-05 – 0.019</td>
</tr>
<tr>
<td>ETHANE, METHANE, N2</td>
<td>1</td>
<td>0.83 – 0.92</td>
<td>1</td>
<td>1</td>
<td>-0.18 – -0.086</td>
<td>0.0074 – 0.022</td>
</tr>
<tr>
<td>All fuels</td>
<td>1</td>
<td>0.85 – 0.96</td>
<td>1</td>
<td>1</td>
<td>-0.16 – -0.041</td>
<td>0.0017 – 0.027</td>
</tr>
</tbody>
</table>

**Radiative Flux**

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ETHANE, HYDROGEN, METHANE, N2</td>
<td>12</td>
<td>0.81 – 0.96</td>
<td>1.1 – 1.4</td>
<td>0.75 – 1</td>
<td>-0.22 – -0.036</td>
<td>0.1 – 0.26</td>
</tr>
<tr>
<td>ETHANE, METHANE, N2</td>
<td>12</td>
<td>0.81 – 1.4</td>
<td>1.1 – 1.3</td>
<td>0.83 – 1</td>
<td>-0.21 – -0.31</td>
<td>0.11 – 0.5</td>
</tr>
<tr>
<td>All fuels</td>
<td>12</td>
<td>0.81 – 1.1</td>
<td>1.1 – 1.3</td>
<td>0.83 – 1</td>
<td>-0.21 – -0.27</td>
<td>0.1 – 0.5</td>
</tr>
</tbody>
</table>
The release of flammable gases, such as Liquefied Natural Gas (LNG) or Liquefied Petroleum Gas (LPG), may result in the formation of a flammable vapour cloud that is often dense, depending on the ambient conditions (particularly wind speed and direction). If this cloud encounters a source of ignition within the surrounding environment, a fire may occur. The characteristics of the resulting fire are dependent upon the release conditions and the environment into which the vapours are released. One test was conducted and the results of the test featuring the use of the passive material (FOAMGLAS® PFS System (Gen 2)) were compared to the baseline data obtained from the reference test.

**Location:** Centro Jovellanos, Asturias, Spain  
**Year:** October 2013  
**Number of cases in series:** 1

### Selection of Tests

- **Available measurements:** Radiative Flux

### Simulation Setup

- **Grids used:** 125 mm

### References

[1] (2014), Vapour & Fire Control Testing of FOAMGLAS PFS System (Gen 2) on LNG
FLACS-CFD 22.2

POSTPROCESS QUANTITIES

Radiative Flux

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHANE</td>
<td>4</td>
<td>0.76</td>
<td>11</td>
<td>1</td>
<td>-0.28</td>
<td>0.094</td>
</tr>
</tbody>
</table>

Geometric mean bias vs Geometric variance

Simulation vs Experiment [kW/m²]

METHANE
10.3 Validation cases

10.3.4.7 Montoir LNG Pool Fire

METHANE

British Petroleum, Elf Aquitaine, Gaz de France, Shell and Total-CFP have collaborated to perform three LNG fire experiments which have been conducted successfully at Montoir, France, under different wind conditions in a shallow 35 m diameter bund. Large tests were essential to provide suitable data for use in models which would reduce uncertainties in the prediction of very large fires. Three fire tests were performed under different wind conditions during 1987.

Location: Gaz de France, Montoir de Bretagne methane terminal, France
Year: 1987
Number of cases in series: 1

SELECTION OF TESTS

Available measurements: Radiative Flux

SIMULATION SETUP

Grids used: 2187 mm

REFERENCES

## Radiative Flux

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>METHANE</td>
<td>3</td>
<td>0.32</td>
<td>4.6</td>
<td>0.33</td>
<td>-1.2</td>
<td>4.6</td>
</tr>
</tbody>
</table>

The diagrams illustrate the geometric variance and the geometric mean bias, as well as the simulation compared to the experiment in kW/m².
To study the various parameters like burning rate, flame emissivity to further understand the amount of heat flux contributed from flames and its variation with respect to different pool diameters. The experimental setup consists of mild steel circular pans of 2 mm thick and 15 cm height. Gasoline with mass burning rate of 77 gm/m2.s is considered as fuel for this study. Heat flux is measured at distance equal to pool diameter away from the pool centre on a vertical axis. The heat flux gauge is traversed along the vertical axis.

Location: INDIAN INSTITUTE OF TECHNOLOGY BOMBAY, India
Year: 2013
Number of cases in series: 1

Available measurements: Radiative Flux

Grids used: 18 mm

### Postprocess Quantities

**Radiative Flux**

<table>
<thead>
<tr>
<th>FUEL</th>
<th>NMGVGFAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>BENZENE, N-_BUTANE, N-_HEXANE, N-_PENTANE, N-_PROPYLENETHANE, O-_XYLENE, TOLUENE</td>
<td>13</td>
<td>1.4</td>
<td>1.2</td>
</tr>
</tbody>
</table>

![Geometric Variance vs Geometric Mean Bias](image1)

![Simulation vs Experiment](image2)

- BENZENE, N-_BUTANE, N-_HEXANE, N-_PENTANE, N-_PROPYLENETHANE, O-_XYLENE, TOLUENE
10.3 Validation cases

Table 10.4: Blast validation cases.

### 10.3.5 Blast

One, medium scale validation case that demonstrates the performance of the FLACS-CFD blast solver are included in the following subsection. Table 10.4 gives a summary of this experiment.
10.3.5.1 Urban Canyon

The objective was to study the blast effects from medium sized high explosive (HE) and thermobaric explosive (TBX) charges detonated in an urban environment. The experiment was scaled 1/5. A simplified urban environment consisting of only four buildings was selected. The buildings were cubic and all dimensions optimized for the purpose of validation of numerical tools. The streets in the urban scenario were chosen to have the same width as the building dimensions, 2.3 meter. Four charge locations and two different heights of burst (0.2, 1.15 m) were used for the explosive charges.

Location: Sweden
Year: 2006
Number of cases in series: 8

Available measurements: Pressure

Grids used: 200 mm

FLACS-CFD 22.2

POSTPROCESS QUANTITIES

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>168</td>
<td>0.89</td>
<td>2</td>
<td>0.85</td>
<td>-0.11</td>
<td>1.5</td>
<td></td>
</tr>
</tbody>
</table>

Geometric mean bias

Simulation [barg]

Experiment [barg]
10.3.6 Dust Explosion

Validation cases that demonstrate the performance of the FLACS-CFD DustEx solver are included in the following subsection. Table 10.5 gives a summary of the dust explosion experiments.

Table 10.5: Validation cases.

<table>
<thead>
<tr>
<th>Case name</th>
<th>Type</th>
<th>Variable</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>Metrics FAC2</th>
<th>FR</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flame Acceleration pipe</td>
<td>Explosion</td>
<td>Pressure</td>
<td>12</td>
<td>1.7</td>
<td>1.4</td>
<td>0.07</td>
<td>0.31</td>
<td>0.36</td>
</tr>
<tr>
<td>1(32)0</td>
<td>Explosion</td>
<td>Pressure</td>
<td>8</td>
<td>1.02</td>
<td>2.6</td>
<td>0.5</td>
<td>-0.79</td>
<td>2.3</td>
</tr>
<tr>
<td>50(2080)</td>
<td>Explosion</td>
<td>Pressure</td>
<td>8</td>
<td>4</td>
<td>3.6</td>
<td>0.25</td>
<td>1</td>
<td>2.1</td>
</tr>
<tr>
<td>coated dust (Hylin0)</td>
<td>Explosion</td>
<td>Pressure</td>
<td>40</td>
<td>0.35</td>
<td>5.9</td>
<td>0.38</td>
<td>-0.31</td>
<td>4.4</td>
</tr>
</tbody>
</table>
The main aim of this experiment is to study the effect of congestion on dust explosion and determination of the fundamental flame characteristics, in a dust-air mixture and the influence of turbulence on flame propagation. The experimental setup consist of tube with diameter of 0.19 m and with two lengths 1.86 m and 0.93 m with both ends closed. Experiments were conducted with and without obstacles within the tube. Obstacle consist of concentric ring with internal diameter of 114 mm and outside diameter of 165 mm with a spacing of 85 mm. 12 such concentric rings are placed for 1.86m tube and 6 for 0.93 m tube. The ignition is done at the center of the bottom of the tube with ignition energy of 1.75 kJ through out the series of tests. Fuel used is corn starch and experiments are conducted at different ignition delay time and with dust concentration varying from 260 g/m³ to 700 g/m³.

Location:  
Year:  
Number of cases in series: 12

Available measurements: Pressure

Grids used: 10 mm

Yi Kang Pu and Jacek Mazurkiewicz and Josef Jarosinski and C. (William Kauffman) (1989), Comparative study of the influence of obstacles on the propagation of dust and gas flames, Symposium (International) on Combustion
FLACS-CFD v22.2 User’s Manual

FLACS-CFD 22.2|670AC58|
DATE: FEBRUAR 06, 2023
POSTPROCESS QUANTITIES

Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>1.9</td>
<td>1.5</td>
<td>0.67</td>
<td>0.57</td>
<td>0.37</td>
<td></td>
</tr>
</tbody>
</table>

![Graph showing geometric mean bias vs geometric variance and experiment vs simulation.]
The experiments were conducted in cylindrical silo of diameter 1.6m and height 5.6m including bin-hopper at the bottom. The silo bottom was filled with sand to prevent the pressure wave from leaving the silo hence the effective volume was then reduced to 9.4 m³. Corn starch dust cloud was generated using different methods; ring nozzles and pressurized dust reservoirs ('homogeneous cloud'), mechanical feeding, pneumatic dust injection tangentially, and pneumatic dust injection vertically downward. Dust concentrations inside conveying pipe of diameter 75mm, was varied as 1, 3, 5, 7 Kg/m³. The silo has vent openings with polyethylene film (0.1 bar). Four vent areas were used in the campaign 0.15m², 0.3m², 0.5m² and 0.7m². The cloud was ignited at three different locations. For the explosion modelling, Silo is assumed to be completely filled with dust cloud.

Location:
Year:
Number of cases in series: 8

Available measurements:
Pressure

Grids used: 100 mm

REFERENCES


FLACS-CFD 22.2

POSTPROCESS QUANTITIES

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0.52</td>
<td>3.6</td>
<td>0.5</td>
<td>-0.79</td>
<td>2.3</td>
<td></td>
</tr>
</tbody>
</table>

- **Geometric mean bias** vs. **Geometric variance**
- **Simulation** vs. **Experiment [barg]**

FLACS-CFD v22.2 User’s Manual
10.3 Validation cases

10.3.6.3 Silo 236m3

SILO 236M3

A experimental silo facility, comprising of 236 m3 steel silo cell of height 22 m and diameter with pneumatic dust injection at bottom and top of the silo. The cloud were ignited at various height above the ground. Two vent area of 5.7 m2 and 3.4 m2 were used.

Location: Bergen, Norway
Year: December, 1985
Number of cases in series: 8

Available measurements: Pressure

SIMULATION SETUP

Grids used: 180 mm

REFERENCES

Skjold, T. and Arntzen, Bjørn and Hansen, Olav and Taraldset, O. and Storvik, Idar and Eckhoff, R. (2005), Simulating Dust Explosions with the First Version of DESC, Process Safety and Environmental Protection
FLACS-CFD 22.2

Postprocess Quantities

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4</td>
<td>16</td>
<td></td>
<td>0.25</td>
<td>1</td>
<td>2.1</td>
</tr>
</tbody>
</table>

![Graphs showing geometric variance and simulation vs experiment data.](image-url)
### 10.3.6.4 Vented Duct 18.5m3

**VENTED DUCT 18P5M3**

Series of dust explosion experiments performed in an 18.5 m³ vessel equipped with vent ducts of varying cross sections and lengths. The effect of vent ducts of larger cross section than the vent are examined and also the effect on the reduced explosion pressure ($P_{red}$) of the location of the ignition source when vent ducts are used.

**Location:** UK  
**Year:** February 1991  
**Number of cases in series:** 60

---

### SELECTION OF TESTS

**Available measurements:** Pressure

---

### SIMULATION SETUP

**Grids used:** 100 mm

---

### REFERENCES


FLACS-CFD 22.2
DATE: FEBRUAR 15, 2023
POSTPROCESS QUANTITIES

Pressure

<table>
<thead>
<tr>
<th>FUEL</th>
<th>N</th>
<th>MG</th>
<th>VG</th>
<th>FAC2</th>
<th>FB</th>
<th>NMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>0.35</td>
<td>5.9</td>
<td>0.38</td>
<td>-0.81</td>
<td>1.4</td>
<td></td>
</tr>
</tbody>
</table>

![Graph showing geometric variance and simulation vs. experiment for pressure data.](image-url)

FLACS-CFD v22.2 User’s Manual
Chapter 11

References

11.1 Bibliography

Below follows the bibliography for this user manual. A complete list of Gexcon's publications can be found online.


Gexcon (2013). Minutes of the FLUG meeting 23–24 May 2013 including the notification “On the use of simulation results obtained with the CFD code FLACS for optimizing safety gaps between repeated congested regions.” Available from the FLUG website.


11.2 Online references


Index

3D output variables, 230
6-flux radiation model, 464
a1file, 358
a3file, 360
abbreviations, 490
ABSCOF (output variable), 235
absorption coefficient, 235
ACH, 357, 492
advanced scenario parameters, 89
aerodynamic roughness length, 122
aggregating results, 320
Air change rate, 357
air leak, 140
ambient occlusion, 306
application areas, 2
area leak, 439
area leak, warnings and errors, 286
arrange items, 204
ASCII data file, 360
ASCII file, 354, 355, 358
autostop, 353
average pressure, 237
axis show, 79
background colour CASD, 190
best practice, 363
beta flame model, 431
BLEVE, 252
boundary conditions, 112
boundary conditions for explosion after dispersion, 118
boundary, Bernoulli, 112
boundary, eqchar, 112
boundary, Euler, 112, 113
boundary, nozzle, 112, 113
boundary, plane wave, 112
boundary, sea surface, 118
boundary, symmetry, 112
boundary, wind, 112, 114, 428
box, 77
canopy height, 122
CASP, 51
casd command line options, 60
CASD error, unable to start, 60
CASD layout manager, 191
cf-file, 111, 269
CFL-numbers file, 111, 270
CFLC, 105
CFLV, 105
CFM, 468
CGNS, 110
CGNS files tools, 361
change output units, 240
change variable name, 240
changelog, 10
changing units, 99
check grid, 85
CHECK_CONVERGENCE, 107
choked flow, 336
clip planes, 203
cloud file, 133, 253
cloud interface, 132, 253
cloud, gas, 123, 132
cn file, 111, 270
CO (output variable), 235
cofile utility, 350, 473
colour scheme, 66
combustion, 185, 462
combustion modelling, 430
comerge, 352
command input CASD, 63
compatibility, 18
conduction, 187, 253, 470
configure wizard, 28
context menu, 291
convective heat flux, 236
convective heat transfer coefficient, 235
coordinate field, 64
coordinates, 64
copy subtree, 76
coupled pool fire physics, 456
coupled pool fires, 402
CP8, 77
cs file, 244
cs.MON file, 270, 283, 451
culling limit, 189
custom gas clouds, 132
cut subtree, 76

gexcon
INDEX

geometry number in job number, 190
geometry selection mode, 64
geometry window, 63
geometry window icons, 64
geometry, large, 205
gexcon colour convention, 66
gm, 331
graphics performance, 190
graphics performance CASD, 190
grating, 165
gravity constant, 119
grid, 82, 331
grid cells, number of, 84
grid display, 87, 188
grid print, 333
grid refine, 334
grid smooth, 87, 334
grid stretch, 87
ground height, 121
ground roughness, 122
ground, 3D plot, 309
GTC, 77
H2O (output variable), 235
hardware requirements, 25
HCONV (output variable), 235
heat dose, 236
heat file, 258
HEAT_SWITCH, 110
heavy hydrocarbons, 288
highlighting while in motion, 190
hinged panel, 95
hinged rigid panel, 96
hinges, 99
HPCON, 251
HPTYP, 251
hue in macro files, 190
humidity, 246, 456
ideal gas law, 424
ignition, 156
import grid, 88
inactive panel, 94, 95
incident radiative flux, 236
incompressible, 110
incompressible solver, 222
inert gas, 128
inert leak, 140
initial conditions, 118
initial temperature, 120
installation, 26
isosurface, 304
jet, 336
jet fire, 288, 391
jet leak, 139
jet sources, 432
jet utility technical description, 434
job number, 9
JOBSPEC namelist, 245
keys, 108
known issues casd, 207
large geometries, 205
layout manager, CASD, 191
leak, 139
leak box, 439
leak buildup time, 140
leak excess area, 286
leak file, 265
leak file, avoid automatic overwriting, 267
leak wizard, 147
left difference, 56, 78
level of details, 189
license server, setting up, 28
license solver, 219
light position, 3D plot, 308
line monitors, 170
list of files, 241
load filter, 65, 189
LOD, 189
louvre panel, 161
macro file, 191
maintenance, 39
mark subtree, 75
mass balance, 425
mass fractions (output variables), 235
mass residual, 250, 284
MASS_RESIDUAL, 249
maximum snap points, 190
memory requirement, 84
mixing height error, 287
mixing of gases, 424
mixture fraction, 423
mixture fraction, transport equation, 425
MODE, 106, 107
modify gas data model, for dispersion, 247
modify gas data model, for explosion, 249
mole fractions, 235
momentum equation, 425
Monin-Obukhov length scale, 428
monitor file, 270, 282
monitor panel, 94, 95
monitor points, 91
move grid line, 87
move subtree, 75
multiple gas clouds, 132
multiple gas monitor regions, 170

news, 10
nomenclature, 487
numerical schemes, 450

oblique jet, problems with, 443
on-grid obstacle, 67
open simulation files, 68
overlay panel, 96
oxidation formation soot model, 469
oxygen fraction in air, 246

P (output variable), 230
P_IMP (output variable), 230
page selection, 293
panel drag coefficient, 98
panel maximum travel distance, 98
panel opening pressure difference, 97
panel porosity, 97
panel pressure, 237
panel sub sizes, 98
panel type imp, 96
panel weight, 97
panels, 95
panels, fire simulations, 99
parallel solver, 220
PARAMETERS namelist, 249
Pasquill class, 121
paste subtree, 76
PDEATH (output variable), 238
PFAC, 251
pipe bundles, 75
plan drawings, 201
plastic panel, 96
plot area, Flowvis, 291
plot linking, 317
plugins, 191
pool depth, 236
pool example, 381
pool fires, 401
pool leakage file, 265
pool setup file, 259
pool setup parameters, 133
pool temperature, 237
pool variables, 236
pool velocity component in x-direction, 237
pool velocity component in y-direction, 237
pool velocity vector, 236
pool volume, 236
pool, how to set Z, 388
POOL_D (output variable), 236
POOL_S (output variable), 236
POOL_T (output variable), 237
POOL_U (output variable), 237
POOL_V (output variable), 237
POOL_VOL (output variable), 236
popout panel, 95
popout rigid panel, 96
Porcalc, 88
porcalc, 190
porosity calculation, 88, 334, 472
porosity calculator, 190
porosity gaps, 472
porosity verification, 298
porosity, set lower limit, 249
position grid line, 86
PP (output variable), 237
PPOR (output variable), 237
PQ (output variable), 237
PQFFDOSE (output variable), 237
PQRADF (output variable), 237
PQTINC (output variable), 237
Prandtl-Schmidt numbers, 427
pressure relief panels, 94
PROBIT (output variable), 238
PROD (output variable), 230
progress variable, 423
project number in job number, 190
PT (output variable), 237
Q (output variable), 236
Q1, 168
Q6, 169
Q7, 169
Q8, 169
Q9, 169
QCONV (output variable), 236
QDOSE (output variable), 236
QRAD (output variable), 235
QRINC (output variable), 236
quick filter, 81
quick grid setup, 82
QWALL (output variable), 236
r1-file, 277
r1file, 354
r3-file, 278
r3file, 355
r3file, line monitor output, 358
radiation, 110, 180, 463
radiation start, 182
radiative heat flux, 235
radiative properties, 468
radiative source term, 235
radiative transfer, 463
INDEX

RADSRCA (output variable), 235
real gas model in the jet utilities, 342
rectangulartorus, 77
redo, 74
result aggregation, 320
rewind-restart, 111, 284
RHO (output variable), 231
rotate subtree, 75
rotated primitives, 190
Rough wall boundary conditions, 428
rt-file, 278, 280
rt.FUEL file, 168
rt.MON file, 282
run script, 218
RunManager, 34, 211
runtime simulation control file, 111, 269
save object, 73
scalar time plot, 293
scalar-time output file, 277
scale subtree, 75
scale subtree in one direction, 75
scenario, 89
scenario conduction, 187
scenario file, 244
scenario settings, 90
scenario Solid Flame Model, 187
script, 218
set font in coordinate field, 64
setup file, 244
SETUP namelist, 245
side outlow, 99
sidebar legend, Flowvis, 318
simulation log files, 278, 280
simulation volume, 86
six-flux radiation model, 464
smoke, 186
snap points, 194
software requirements, 25
Solid Flame Model, 187
soot, 186, 468
SOOT (output variable), 235
species, 171
speed of sound, 424
starting Flacs simulations, 217
status bar, 65
status field, 64
steady-state solver, 225
stoichiometric mixture, 423
stoichiometric reaction, 424
stop criteria, 102
stopping simulations, 218
subgrid obstacle, 67
suction, 140

T (output variable), 231
TDOSE (output variable), 238
terrain import, 195
terrain roughness, 365
tf_leak, 437
thermal active walls, 253, 470
time slider, Flowvis, 292
TMAX, 102
TMIN, 101
TMOLE (output variable), 238
tool bar, Flowvis, 290
tools, 194
TORUS, 77
TOT_GASkg, 168
total heat flux, 236
toxic dose, 238
toxic mole fraction, 126, 238
toxic probability of death, 238
toxic probit function, 238
toxic substances, 238
tracer, 117
translate primitive, 74
translate subtree, 75
tt-file, 280
TURB (output variable), 231
TURBI (output variable), 231
turbulence model, 426
turbulence viscosity, 426
turbulent kinetic energy, transport equation, 426
TWALL (output variable), 234
two-phase flow, 348
underlay, 201
underpredicting in empty enclosures, 288
UNDERRELAX "Underrelaxation P and UVW", 107
undo, 74
union, 76
unspecified panel, 95
user-defined compound variable, 361
user-defined species, 171
utilities, 327
validation, 493
variables, 491
variables for fire simulations, 234
variables for pool simulations, 236
velocity, friction, 429
ventilation time stepping, 247
version history, 23
vessel burst, 250
VFCO2 (output variable), 235
VFH2O (output variable), 235
VFSOOT (output variable), 235
Visibility (output variable), 235
volume blocked by geometry, 278
volume fractions, 235
volumetric visualisation, 304
VVEC (output variable), 232

wall functions, 427
wall heat impact, 236
wall temperature, 234
warning, 284
water spray, 158
water spray nozzle type, 160
weighted sum of gray gas model, 468
wind, 121
WIND boundary condition, 114, 116, 118, 373, 393, 415, 428
wind buildup time, 116
wind direction, 116
wind speed, 115
wind wizard, 114
WSGGM, 468